NCPA PROPAGATION CODE USERS MANUAL

By

Andrew T. Winkelman

APPROVED:

Dr. John V. Olson

Dr. David E. Fee

Dr. Curt A. Szuberla
Advisory Committee Chair

Dr. Renate A. Wackerbauer
Chair, Department of Physics

NCPA PROPAGATION CODE USERS MANUAL

A

PROJECT

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

By

Andrew T. Winkelman, B.S.

Fairbanks, Alaska

December 2015

Table of Contents

List of Figures

List of Appendices

Preface

This manual was written for University of Alaska Fairbanks infrasound group to assist researchers in using the National Center for Physical Acoustics (NCPA) code suite to further investigate observed infrasonic phenomena. The NCPA code suite is designed to simulate various aspects of infrasound propagation through a model atmosphere. This suite was developed and tested by the University of Mississippi National Center for Physical Acoustics infrasound group. Included are raytrace routines to initially establish signal paths, both single frequency and broadband modal routines to calculate pressure fields and transmission losses, and a parabolic method to calculate pressure fields and transmission losses in model atmospheres.

Acknowledgements

Thanks to Dr. Curt A. Szuberla, Dr. Dave Fee, and Dr. John Olson at the University of Alaska Fairbanks Geophysical Institute for serving as my graduate advisory committee, to Dr. Roger Waxler and Claus Hetzer at the University of Mississippi National Center for Physical Acoustics for assistance with compilation and usage of the NCPA propagation code suite, and to Dr. Doug Drob at the U.S. Naval Research Laboratory for assistance with the Ground to Space (G2S) atmosphere specification.

Chapter 1 Introduction

In acoustic and seismic studies, propagation simulations are a common research method. In infrasound studies, established simulations codes are often difficult to come by and are usually developed by individuals to investigate a particular situation; each individual must spend time and effort developing their own code. This standardized, though proprietary, set of propagation codes for infrasound studies has been developed by the National Center for Physical Acoustics (NCPA) at the University of Mississippi and provided to the infrasound researchers here at UAF. We will discuss the components, requirements, common usage, and limitations of the NCPA Propagation Code package, effectively training researchers in operation of these 'off the shelf' routines. In addition to general information, we will compare results from the NCPA Propagation Code package to a previously published case study: regional-scale propagation of infrasound through moderately strong stratospheric wind jets observed during the Sayarim calibration experiments in 2009.

This section outlines the requirements and conventions used in this user's manual. Section 2 contains instructions for compilation of the code and outlines the operation of the routines. Section 3 explains the physics considered in the development of the routines. Finally, Section 3.4 describes the usage of the routines in a case study situation similar to one a researcher would be interested in. Further information and examples are available in the appendices.

1.1 Required Knowledge

This manual assumes one has basic understanding of maneuvering and manipulating files and folders in the bash command line environment provided by the Linux terminal. While understanding of the intricacies of bash commands is not required, there are many tutorials available online if one desires to know more. Further, while the NCPA software is written in C++ and FORTRAN, it is only required that the user understand the basic mechanics of compilation as all instructions are provided in Section 2.2. Additionally, examples of plotting data will utilize GNUplot. Example GNUplot scripts are provided in Appendix 3.

1.2 Conventions

Bash terminal commands will be presented within this manual. These should be typed verbatim or by substituting relevant fields or values. To denote a terminal command, a characteristic symbol will be placed preceding each command: a bash command will have a "$" symbol preceding the command; e.g.,

```
$ echo 'Hello, World!'
```

For reference, example Bash wrapper scripts used to call the NCPA routines are available in Appendix 2.

GNUplot will be used to produce plots of data. Note that plotting may be done in a gunplot terminal or directly from the bash command line, depending on your system and configuration of gunplot. Presented within this manual are GNUplot scripts or commands passed from the bash terminal into the GNUplot environment using the "`-e`" flag. The usual GNUplot terminal options used to produce plots within this manual are:

```
set terminal png size 1280,640 background rgbcolor "white"
```

Additionally, examples of GNUplot scripts to produce plots presented in this manual are available in Appendix 3.

Some atmospheric data manipulation was performed in MATLAB, though the same manipulations can be done in many other languages. These example scripts are presented for reference in Appendix 4.

2.1 Dependencies

The NCPA code is written for use on Linux systems (it may be compiled on OS X systems prior to version 10.9 "Mavericks" but this manual will not cover such use). For the development of this manual, Ubuntu versions 14.04, 14.10, and 15.04 were used. Of course, several dependencies must be met before compilation. The most reliable method to install dependencies is via a package manager such as `apt-get` but each can be compiled from source if so desired. The dependencies required are listed here with recommended versions:

`fftw 3.3.3` Discrete Fast Fourier Transform subroutine library for C.
`http://www.fftw.org/`

`gcc` Compiler package provides both g++ and gfortran and is usually already installed in most Linux distributions. For OS X, g++ is best obtained through XCode; obtain gfortran separately.

`gfortran 4.7` GNU Fortran complier for GCC.
`https://gcc.gnu.org/wiki/GFortran`

`gnuplot 4.6.3` Portable command-line driven graphing utility. On OS X, ensure x11 is available and support is enabled during gnuplot installation.
`http://www.gnuplot.info/`

`gsl 1.15` GNU Scientific Library, provides many more mathematical functions for C.
`http://www.gnu.org/software/gsl/`

`open-mpi 1.6.5` High performance message passing library that assists in optimizing computations through thread parallelization and concurrency.
`http://www.open-mpi.org/`

`petsc 3.2-p5` A suite of data structures and routines for parallel solution of scientific applications modeled by PDE. Note: installation will be concurrent with compiling the NCPA code in Section 2.2.
`http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.2-p7.tar.gz`

`slepc 3.2-p3` Scalable library for eigenvalue problem computations. Note: installation will be concurrent with compiling the NCPA code in Section 2.2.
`http://slepc.upv.es/download/download.php?filename=slepc-3.2-p5.tar.gz`

`X11` The X11 window system often already installed on Linux and older OS X systems. For OS X 10.6 or later, use XQuartz available at `http://xquartz.macosforge.org/landing/`.

2.2 Installation

Installation is done by the standard *nix compilation procedure. First, unzip the NCPA archive to your desired directory. It is important that the directory be one you have read/write access to. This directory will be denoted `$ncpaprop`. The tar command works well:

```
$ tar zxvf ncpaprop.tar.gz
```

Note that the `petsc` and `slepc` packages do not need to be unzipped and installed. These should be placed in the `$ncpaprop/src/extern` directory; the makefile will unzip and compile these.

Once these dependencies are installed, compilation of the NCPA code can be a time-intensive task. Both the initial release (`v 1.0.0`) and the updated 'work-in-progress' version `WIP20130717` have been shown to compile and function smoothly. The `WIP20130717` includes many complex (read: $a + bi$) eigenvalue versions of the normal mode analysis routines. Troubleshooting tips are presented in Appendix A.

Compile the NCPA code itself. Run:

```
$ cd $ncpaprop/src
$ make
```

If no errors result, ensure the presence of executable binary files in the directory `$ncpaprop/bin`.

2.3 Testing

Next, ensure proper operation of each routine. Quick, basic tests can be run using the samples data files provided (more on provided samples in Section 4.3). Note that during the execution of most routines in the software a plot is automatically generated by gnuplot using an X-window terminal; ensure this functionality is available and enabled on OS X 10.6 and systems and later (see Section 2.1). Testing of each routine is done by first navigating to `$ncpaprop/samples` and running simple cases of each routine. Most routines automatically generate a plot of relevant results at the end of execution. Example scripts for such tests and expected outputs are available in Appendix B.

2.4 The Routines

The software's routines are listed here with a basic description of function and requirements. More in-depth physical considerations are discussed in Section 3.

2.4.1 Raytrace in 2D & 3D

Given an atmospheric specification, the path of infrasound emanating from a sound source or event and transmission losses due to atmospheric attenuation are calculated.

Standard geometrical acoustic methods are employed by iterating Snell's Law of refraction: "The elevation angle $\gamma(z) = \arctan(\mathrm{d}z/\mathrm{d}x)$ and speed c of a plane wave varies with height z such that the ratio $\cos(\gamma(z))/c(z)$ is constant." [10]

2.4.2 ModESS & CModESS

`ModESS` is a range-independent infrasound propagation algorithm in a stratified atmosphere based on the effective sound speed approximation. This approximation is valid for near-horizontal propagation angles and relatively low wind speeds. ModESS and CModESS solve the Helmholtz equation in two dimensions for a point source in a stratified atmosphere with horizontal wind over rigid ground:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \rho_0(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho_0(z)}\frac{\partial}{\partial z}\right) + \frac{\omega^2}{c_{eff}^2(z)}\right] p(r,z) = -\frac{\delta(r)\delta(z-z_s)}{2\pi r}, \tag{2.1}$$

where $p(r,z)$ is the pressure at range r and height z and $c_{eft}(z)$ is the effective (complex) sound speed. This sound speed is the sum of the sound speed and the horizontal component of wind velocity in the direction of propagation, $\hat{k_r}$:

$$c_{eff}(z) = c(z) + \vec{v_0}(z)\cdot\hat{k_r} + ia(w), \tag{2.2}$$

where the ia term accounts for atmospheric absorption and damping of acoustic energy. The boundary conditions are that the vertical change in pressure at the ground is zero (rigid ground) and the vertical change at the maximum height is zero (rigid top, usually greater than 150 km).

`CModESS` is the solves the same equation but the attenuation in the atmosphere is treated exactly instead of perturbatively. As a consequence, the eigenfunctions and eigenvalues are complex. Note: *Because complex eigenvalue numerical methods are employed, convergence is not always guaranteed.*

2.4.3 ModESSrd1wcm & ModESSrd2wcm

`ModESSrd1wcm` and `ModESSrd2wcm` are range-dependent propagation algorithms featuring the same effective sound speed approximation as the previous `ModESS` algorithms. The attenuation in the atmosphere is taken into account as a perturbation. These algorithms solve the Helmholtz equation in two dimensions for a point source in a stratified atmosphere with (local) horizontal wind over rigid ground:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \rho_0(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho_0(z)}\frac{\partial}{\partial z}\right) + \frac{\omega^2}{c_{eff}^2(z)}\right] p(r,z) = -0, \tag{2.3}$$

where the point source is placed in the first range segment and has the form

$$-\frac{\delta(r)\delta(z-z_s)}{2\pi r}. \tag{2.4}$$

2.4.4 WMod

`WMod` is a range-independent, wide-angle, high-Mach, modal code to simulate infrasound propagation in a stratified atmosphere. This routine can handle significantly wider propagation angles and higher wind speeds than `ModESS`. WMod solves the Helmholtz equation in 2 dimensions for a point source in a stratified atmosphere with horizontal wind over rigid ground:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\rho_0(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho_0(z)}\frac{\partial}{\partial z}\right)+\frac{(\omega+i\vec{v_0}\cdot\nabla_r)^2}{c^2(z)}\right]p(r,z)=-\frac{\delta(r)\delta(z-z_0)}{2\pi r}. \tag{2.5}$$

Note the lack of effective sound speed approximation.

2.4.5 ModBB & CModBB

`ModBB` propagates an infrasound pulse with a given bandwidth and center frequency is propagated to a specified distance in a range-independent stratified atmosphere. `ModBB` uses either the `ModESS` or `WMod` routines to propagate the pulse, depending on the Mach number.

`CModBB` is the solves the same equation but the attenuation in the atmosphere is treated exactly instead of perturbatively. As a consequence, the eigenfunctions and eigenvalues are complex. Note: *Because complex eigenvalue numerical methods are employed, convergence is not always guaranteed.*

2.4.6 PaPE: High order parabolic equation calculation

`PaPE` uses the Parabolic Equation method to solve for infrasound propagation in a range-dependent stratified atmosphere over rigid ground. The code solves the Helmholtz equation in two dimensions:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\rho_0(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho_0(z)}\frac{\partial}{\partial z}\right)+\frac{(\omega+i\vec{v_0}\cdot\nabla_r)^2}{c^2(z)}\right]p(r,z)=-\frac{\delta(r)\delta(z-z_0)}{2\pi r}. \tag{2.6}$$

2.5 Running the Routines

These routines are run from the command line with options and parameters input via the usual flag method. A list of all options is available in the built in help file accessed by running each routine without options or with the help flag, e.g.

```
$ ./raytrace.2d --help
```

The help output is also presented in Appendix C.

All routines require information about the atmosphere in order to function. Examples of sufficient atmosphere specifications include the Naval Research Laboratory Ground to Space semi-empirical spectral model (commonly called a "G2S" profile)[4] or a space-delimited, column formatted ASCII data file. For range independent studies, a single column-based ASCII data file is sufficient; for range dependent studies, a set of ASCII files or a G2S file can be used. If an ASCII file is used, the `--atmosfile` flag is used to point to the atmosphere profile and the `--atmosfileorder` flag is used to denote the order of columns within the atmosphere profile. The `--skiplines` flag is used to designate a number of header lines at the beginning of the atmosphere file to ignore.

The raytrace routines require information specifying rays to trace in addition to an atmosphere profile. The `--azimuth`, `--elev`, `--maxraylength`, `--maxheight`, and `--skips` flags can be useful to restrict investigation to desired ray candidates. See the help file for descriptions and full list of options.

The single-frequency modal routines require input similar to the raytrace routines with the addition of frequency, denoted though the use of the `--freq` flag. These routines can be used to calculate 2D transmission loss by using the `--write_2D_TLoss` flag and can calculate 2D transmission loss for various azimuths by using the `--Nby2Dprop` flag. Again, see the help file descriptions and full list of options.

The broadband modal routine uses the single-frequency modal routines iteratively to construct a final propagated waveform. The single-frequency algorithm is chosen by using the `--use_modess` or `--use_wmod` flags. To propagate a pulse, a dispersion file must be created before propagation the actual pulse. This involves running ModBB twice with various options; these are described in full within the help file.

The parabolic equation routine can utilize a 1D atmosphere profile, a set of atmosphere files forming a 2D profile by using the `--use_1D_profiles_from_dir` flag pointing to a directory of ASCII files, or a G2S environment file by using the `--g2senvfile` flag and providing a file path and name. The remaining functions are similar to those in the modal routines and are described and listed in the help file.

Note that it is useful to run each routine from a script rather than from the command line directly. This allows options and parameter settings to be preserved for another time and is a convenient place to organize output files. Simply put the command, e.g. the tests in Section 2.3, in a text file and ensure its file permissions allow execution. Examples of such scripts are provided in Appendix 2.

Chapter 3 Physical Considerations: Propagation & Dynamics

The routines in the NCPA suite solve modified Helmholtz equations built from the linearized acoustic equations. Section 3.1 discusses the physics and derives the model equations governing sound propagation using these linearized equations. Some effects on propagation resulting from various constructs in an atmosphere, such as wind, are discussed; observable phenomena such as refraction and reflection are explored as well. Physical considerations in simulating sound propagation by both normal mode and parabolic methods are discussed in Section 3.4.

3.1 Derivations of Basic Equations

3.1.1 Linear Acoustic Equations

A simple atmosphere is assumed to be a compressible and relatively homogeneous fluid with sound tending to cause local pressure, density, and fluid velocity perturbations. As such, consider small local perturbations to the background pressure, density, and velocity in the conservation of mass and momentum equations[10]

$$\text{Mass conservation: } \frac{\partial \rho_a}{\partial t} + \vec{\nabla} \cdot (\rho_a \vec{v}_a) = 0 \tag{3.1}$$

$$\text{Momentum conservation: } \rho_a \frac{\partial \vec{v}_a}{\partial t} + \vec{\nabla} p_a = 0, \tag{3.2}$$

where p_a is the ambient pressure in the atmosphere, ρ_a is the atmospheric density, and $\vec{v}_a$ is the flow velocity in the atmosphere. The subscript a means "atmosphere" implying both the background and fluctuation are included, i.e.,

$$p_a = p_{\text{avg}} + p \tag{3.3}$$

$$\rho_a = \rho_{\text{avg}} + \rho \tag{3.4}$$

$$\vec{v}_a = \vec{v}_{\text{avg}} + \vec{v}, \tag{3.5}$$

where p, ρ, and $\vec{v}$ are the small acoustic fluctuations compared to the average background p_{avg}, ρ_{avg}, and $\vec{v}_{\text{avg}}$. Even for a highly impulsive event, e.g., an explosion, the resulting pressure wave can be on the order of a 10^3 Pa while 1 atmosphere is on the order of 10^6 Pa; the ratio is still small on the order of 10^{-3}. The assumed homogeneous character of the atmosphere means that spatial and temporal derivatives of the background terms are zero. Additionally, we consider an atmosphere with no background wind, meaning $\vec{v}_{\text{avg}} = 0$. Expanding the conservation equations with these small acoustic fluctuations and linearizing results in the following linear acoustic equations yields:

$$\frac{\partial \rho}{\partial t} + \rho_{\text{avg}} \vec{\nabla} \cdot \vec{v} = 0 \tag{3.6}$$

$$\rho_{\text{avg}} \frac{\partial \vec{v}}{\partial t} + \vec{\nabla} p = 0. \tag{3.7}$$

3.1.2 Wave Equation

The thermodynamic properties of the small perturbations are assumed to be adiabatic: no heat flow results from this process[10]. Given this, pressure and density are related by

$$p_a = K \rho_a^{\gamma}, \tag{3.8}$$

with K a constant and $\gamma = \frac{c_p}{c_v}$ the specific heat ratio. Including the small perturbations and Taylor expanding, pressure and density fluctuations are then related by

$$p = \left(\gamma \frac{p_{\text{avg}}}{\rho_{\text{avg}}} \right) \rho = c^2 p, \tag{3.9}$$

with c the adiabatic sound speed.

With this new relation, the density perturbation, ρ, can be eliminated from the linear acoustic equations

$$\rho_{\text{avg}} \vec{\nabla} \cdot \vec{v} + \frac{1}{c^2} \frac{\partial p}{\partial p} = 0 \tag{3.10}$$

$$\rho_{\text{avg}} \frac{\partial \vec{v}}{\partial t} + \vec{\nabla} p = 0. \tag{3.11}$$

Then, taking a time derivative of the first acoustic equation and a space derivative of the second and eliminating the resulting $\vec{\nabla} \cdot \frac{\partial \vec{v}}{\partial t}$ term results in the linear wave equation:

$$\vec{\nabla}^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0. \tag{3.12}$$

This is the wave equation governing propagation of a pressure wave through a homogeneous atmosphere.

3.1.3 Helmholtz Equation

Consider a pressure wave that oscillates sinusoidally with time (this means that local density and velocity fluctuations follow suit)[10]; i.e.,

$$p = A \cos\left(\phi - \omega t\right), \tag{3.13}$$

where A is the sound field amplitude, ϕ is a phase factor, and ω is the frequency of oscillation in time. For this harmonic sound, the wave equation becomes

$$\vec{\nabla}^2 p + k^2 p = 0, \tag{3.14}$$

with $k = \frac{\omega}{c}$ is the wave number. This relation is known as the Helmholtz equation.

3.2 Dispersion Relation

A derivation of the dispersion relation for sound waves in the atmosphere was presented in Hines[8] and in Bodily's master's project[2], ignoring dissipation terms. Beginning from the following set of equations

$$\frac{\partial P}{\partial t} + \vec{v} \cdot \vec{\nabla} P = \frac{\gamma P}{\rho} \left(\frac{\partial \rho}{\partial t} + \vec{v} \cdot \vec{\nabla} \rho \right), \tag{3.15}$$

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}) = 0, \tag{3.16}$$

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} \right) + \vec{\nabla} P - \rho \vec{g} = 0, \tag{3.17}$$

$$P = P_0 \, \mathrm{e}^{-z/H}, \tag{3.18}$$

$$H = \frac{c^2}{\gamma g}, \tag{3.19}$$

where P is the atmospheric pressure, $\vec{u}$ the local velocity of an air parcel, γ the specific heat ratio, ρ is the mass density, $\vec{g}$ the acceleration due to gravity, H the pressure scale height, and c the sound speed.

Then, consider small perturbations in total pressure, density, and velocity (and setting the background wind velocity to zero),

$$P = P_0 + \bar{P}, \tag{3.20}$$

$$\rho = \rho_0 + \bar{\rho}, \tag{3.21}$$

$$\vec{u} = \vec{\bar{u}}, \tag{3.22}$$

with the subscript zeroes indicating the background quantities unperturbed by any waves and the bar indicating a small acoustic fluctuation. Substituting these fluctuation expressions into the beginning set of equations (3.15)-(3.19) and taking Fourier expansions in time and a horizontal spatial coordinate (i.e. $\partial/\partial t \to -i\omega$, $\partial/\partial x \to ik$) results in the following dispersion relation

$$\omega^4 - \omega^2 c^2 \left(k_x^2 + k_z^2 \right) + (\gamma - 1) \, g^2 k_x^2 - \frac{\gamma g^2 \omega^2}{4c^2} = 0, \tag{3.23}$$

where ω is the wave frequency, c the sound speed, k_x the horizontal wave number, k_z the real part of the vertical wave number, γ the ratio of specific heats, and g the acceleration due to gravity.

This dispersion relation allows the definition of the Brunt-Vaisala frequency, ω_b,

$$\omega_b^2 = \frac{g^2 (\gamma - 1)}{c^2} \tag{3.24}$$

and the acoustic cutoff frequency, ω_a,

$$\omega_a^2 = \frac{g^2 \gamma^2}{4c^2}. \tag{3.25}$$

With these defined, the dispersion relation becomes

$$\omega^4 - \omega^2 c^2 \left(k_x^2 + k_z^2\right) + \omega_b^2 c^2 k_x^2 - \omega_a^2 \omega^2 = 0. \tag{3.26}$$

3.2.1 Effective Sound Speed

There are some assumptions that were made during the development of the wave and Helmholtz equations which may not be valid in all cases. A real atmosphere is certainly not homogeneous and exhibits a background wind. A background wind has the effect of transporting the medium carrying the sound pressure perturbations. Thus, we develop the effective sound speed.

Consider a Galilean-style transform, where the component of the wind in the direction of propagation is added to the adiabatic sound speed [10]

$$c_{\text{eff}} = c + \vec{v}_{\text{avg}} \cdot \hat{k}_r, \tag{3.27}$$

where $\hat{k}_r$ is the unit vector in the direction of horizontal propagation. The Helmholtz equation is then written using the effective wave number

$$\vec{\nabla} p + \left(\frac{\omega}{c_{\text{eff}}}\right)^2 p = \vec{\nabla} p + (k_{\text{eff}})^2 p = 0. \tag{3.28}$$

3.3 Atmospheric Absorption

Atmospheric properties, such as viscosity and scattering, contribute to dissipative losses. These losses depend on the frequency of the sound propagated and on features of the atmosphere, such as temperature, density and humidity differences. These effects have roughly an exponentially decreasing effect on sound amplitude with propagated range. As discussed by Salomons [10], atmospheric absorption has an effect on the phase speed of a sound wave too, which tends to disperse a broadband wave. These effects can be

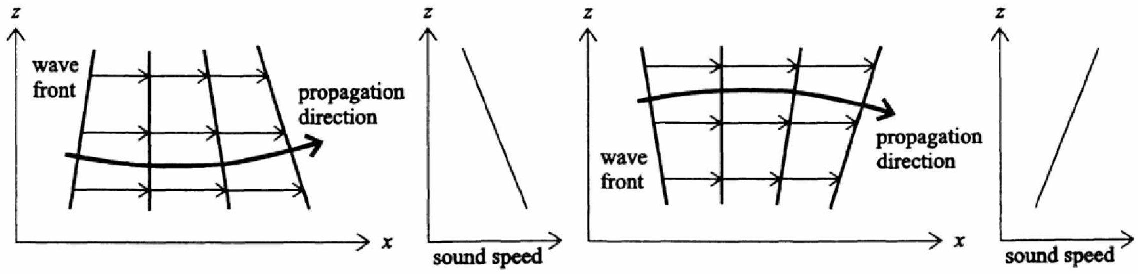

Figure 3.1: Refracting atmosphere visualization; left plot is an example of an upward refracting atmosphere while the right plot is a downward propagating atmosphere. In this visualization using plane waves, the difference in phase speed with height causes a net change in propagation direction. Image credit [10].

implemented by including a small correction terms to the wave number

$$k_{\text{absorb}} = k + i\beta, \tag{3.29}$$

where $\beta = \alpha/20 \log e$ and α is the absorption coefficient in dB per unit length. This leads to the pressure amplitude having an exponentially decreasing quality

$$p = A\mathrm{e}^{-s\beta} \cos(\phi - \omega t) . \tag{3.30}$$

For most practical applications, phase effects of atmospheric absorption can be neglected.

An improved standard atmospheric absorption model has been developed by Sutherland and Bass [11] and used by default in the NCPA Propagation code. This model is designed specifically for infrasound studies and is used to predict suitable absorption coefficients up to 160 km.

3.3.1 Layered Atmospheres

With the homogeneous atmosphere discussed previously, no curvilinear ray path propagation can occur. This result is the simplified case of the linearized acoustic equations and homogeneous Helmholtz equation. In such a simple system, effects such as refraction will not be seen. Refraction occurs when there is a change in sound propagation direction due to a difference in sound speed with altitude. Such a refracting atmosphere will guide a sound pressure wave upward or downward, exhibited by an upward or downward refracting atmosphere, respectively. This effect is easily pictured in Figure 3.1 by considering snapshots of a propagating plane wave front and considering the net propagation direction, defined by the vector normal to the plane wavefronts at each point.

For such a layered atmosphere, considerations of the changing atmospheric properties along the vertical direction need to be taken into account when deriving the acoustic and Helmholtz equations. Salomons [10] has a concise derivation of the homogeneous Helmholtz equation in a layered atmosphere, given by

$$\rho \vec{\nabla} \cdot \left(\frac{1}{\rho} \vec{\nabla} p \right) + k^2 p = 0. \tag{3.31}$$

3.4 Computational Methods

Three principle methods are employed within the NCPA suite routines: raytracing, normal-mode methods, and a parabolic equation method.

In general, the raytrace routines are based on an iterated numerical integration of Snell's law of refraction, which states, "the elevation angle $\gamma(z) = \arctan(dz/dx)$ and speed c of a plane wave varies with height z such that the ratio cos $\gamma(z)/c(z)$ is constant" [10]. This is a geometrical acoustic approach.

The modal routines, such as `ModESS` and `Wmod`, use a normal mode approach such as the method presented by Jensen[9]. Beginning with the two dimensional, cylindrically symmetric Helmholtz equation with a point source and density and sound speed depending only on height,

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial p}{\partial r} \right) + \rho(z) \frac{\partial}{\partial z} \left(\frac{1}{\rho(z)} \frac{\partial p}{\partial z} \right) + \frac{\omega^2}{c^2(z)} p = -\frac{\delta(r)\delta(z)}{2\pi r}. \tag{3.32}$$

A separable solution of the unforced equation is given by the form $p(r, z) = \Phi(r)\Psi(z)$. After substituting this ansatz and dividing,

$$\frac{1}{\Phi} \left[\frac{1}{r} \frac{d}{dr} \left(r \frac{d\Phi}{dr} \right) \right] + \frac{1}{\Psi} \left[\rho(z) \frac{d}{dz} \left(\frac{1}{\rho(z)} \frac{d\Psi}{dz} \right) + \frac{\omega^2}{c^2(z)} \Psi \right] = 0. \tag{3.33}$$

Following with the separation of variables method, the quantities in brackets are functions of only z and r, respectively. Denoting the separation constant as k_{rm}^2 and the particular functions for each separation constant as $\Psi_m(z)$, the modal equation is

$$\rho(z) \frac{d}{dz} \left[\frac{1}{\rho(z)} \frac{d}{dz} \right] \Psi_m(z) + \left[\frac{\omega^2}{c^2(z)} - k_{rm}^2 \right] \Psi_m(z) = 0, \tag{3.34}$$

with

$$\Psi(0) = 0, \text{ and } \left. \frac{d\Psi}{dz} \right|_{z=D} = 0. \tag{3.35}$$

This modal equation is a typical Sturm-Liouville eigenvalue problem, the solutions to which have many useful properties including that the modes form a complete orthonormal set. This can be used to form an expression for the complex pressure field $p(r, z)$ in the form of a Hankel function of the first kind,

$$p(r, z) \simeq \frac{i}{\rho(z_s)\sqrt{8\pi r}} \mathrm{e}^{-i\pi/4} \sum_{m=1}^{\infty} \Psi_m(z_s)\Psi(z) \frac{\mathrm{e}^{ik_{rm} r}}{\sqrt{k_{rm}}}, \tag{3.36}$$

using the asymptotic approximation to the Hankel function. The transmission loss can then be calculated,

$$\mathrm{TL}(r,z) = -20\log\left|\frac{1}{\rho(z)}\sqrt{\frac{2\pi}{r}}\sum_{m=1}^{\infty}\Psi_m(z_s)\Psi_m(z)\,\frac{\mathrm{e}^{ik_{rm}r}}{\sqrt{k_{rm}}}\right|. \tag{3.37}$$

The parabolic equation method, `PaPE`, is most useful for solving range-dependent propagation problems. A standard way to arrive at a 2D parabolic wave equation is presented in Jensen[9] and follows. Beginning with a standard 2D Helmholtz equation,

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2} + k_0^2 n^2 p = 0, \tag{3.38}$$

where $p(r,z)$ is the acoustic pressure, $k_0 = \omega/c_0$ a reference wavenumber, and $n(r,z) = c_0/c(r,z)$ the index of refraction. Then, the solution takes the form

$$p(r,z) = \psi(r,z)\,H_0^{(1)}\left(k_0 r\right), \tag{3.39}$$

which is similar to that of the normal modes method but with $\psi(r,z)$ assumed to be slowly varying in range. Substituting this trial solution and making a farfield assumption that $k_0 r \gg 1$, a simplified elliptic wave equation results,

$$\frac{\partial^2 \psi}{\partial r^2} + 2ik_0\frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} + k_0^2\left(n^2 - 1\right)\psi = 0. \tag{3.40}$$

Then, using the small angle or paraxial approximation,

$$\frac{\partial^2 \psi}{\partial r^2} \ll 2ik_0\frac{\partial \psi}{\partial r}, \tag{3.41}$$

the standard parabolic equation is obtained,

$$2ik_0\frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} + k_0^2\left(n^2 - 1\right)\psi = 0. \tag{3.42}$$

Solutions for ψ and thus acoustic pressure are then sought, followed by a transmission loss calculation similar to the expression in Equation (3.37) for the normal mode method.

Chapter 4 Preparing A Case Study

4.1 Which Routine to Run

Deciding which routine to run depends on the information sought. Is a range independent calculation sufficient or is range dependence required? Are 2D Transmission loss calculations required or is 1D enough; should an 2D pressure field be calculated? Is the wind fast enough to require high-mach considerations? Is the situation in question involving a sufficiently “loud” source to require a parabolic calculation? These are all considerations that should be made before approaching the terminal.

Often, the best place to begin is with a 2D raytrace. Raytrace routines are computationally inexpensive methods to geometrically determine the possibility of a signal arrival before other, more involved, methods are employed. The `raytrace.2d` routine is easily configure to use a single column formatted, ASCII atmosphere profile such as the provided NCPA Canonical Atmosphere template (see Section 4.3) for range independent calculations. This routine can also be iterated to calculate a set of rays with different launch elevations or different azimuths. The 3D raytrace routine, `raytrace.3d`, is used similarly to `raytrace.2d` and can be used to investigate horizontal deviation.

The next step is to investigate using single frequency, range-independent, normal mode routines. If the model atmosphere does not include high winds and launch elevation angles under 30 degrees are of interest, then the faster `Modess` routine should be used. For high wind situations or high launch elevation investigations, the slower but more accurate `WMod` routine should be used. Both routines can accept a single column formatted ASCII atmosphere profile for range independent calculations and can be configured to output both 1D and 2D transmission loss results. It is recommended to ensure proper execution of `Modess` or `WMod` single frequency calculations before moving on to broadband investigations.

For broadband calculations, `ModBB` should be used. This routine iterates the single frequency `Modess` or `WMod` routines to calculate transmission losses for a discrete range of frequencies and calculates a set of dispersion files for each operating frequency. `ModBB` is then run a second time to propagate an initial wave or pulse through the provided model atmosphere using the previously calculated dispersion information. This routine similarly accepts a column formatted ASCII atmosphere profile and as such is also only a range independent calculation.

Finally, `PaPE`, the parabolic equation based routine, is capable of range dependent calculations. This routine can accept a G2S environment file or a set of extracted ASCII profiles for range dependent calculations or a single ASCII profile for range independent calculations. The number of coefficients (number of terms to consider in the parabolic method) can be adjusted for accuracy. 1D and 2D transmission loss is calculated

4.2 Atmospheric Data

Atmospheric specification is required for the NCPA routines to function and is a critical component of any atmospheric acoustic propagation study. Atmospheric specifications can be obtained from a variety of sources depending on the altitude of interest. For low altitude atmospheric specification, data from radiosonde launches twice a day at most major airfields can be used. Radiosonde data is generally reliable up to 35 to 40 km altitude. One source of radiosonde data is the University of Wyoming's archive at `http://weather.uwyo.edu/upperair/sounding.html`. For higher altitude data, seek repositories of infrared or microwave sounding measurements, LIDAR, RADAR, or even satellite-based measurements. Specification from different atmospheric measurements can be concatenated together to form a larger or more complete profile as long as all required data is present. The NCPA routines generally accept an atmospheric specification in the form of a space-delimited ASCII 'ZUVWTDP' (named for the entries in each column) data file. The data columns needed for the ZUVWTDP profile are:

- Z: vertical height in kilometers above sea level;
- U: zonal wind speed in meters per second;
- V: meridional wind speed in meters per second;
- W: vertical wind speed in meters per second (zero; current versions of the code ignore this column);
- T: temperature in Kelvin;
- D: air density in grams per cubic centimeter;
- P: ambient pressure in hectoPascals.

Take care in calculating some of these quantities, especially wind speeds, as standard practices in meteorology are to record the direction *from* which the wind is blowing; the expected format is a vector decomposition of horizontal wind velocity using the South to North and East to West axes. Note also that air density calculations should be done considering partial pressures of humid and dry air[7]. The total pressure recorded, p, is the sum of partial pressures from dry and humid air: $p = p_d + p_v$, where p_d is the dry air pressure and p_v is the water vapor pressure. Water vapor partial pressure is calculated from the relative humidity as $p_v = \phi p_{sat}$, with ϕ the relative humidity and p_{sat} the water vapor saturation pressure, measured in hectoPascals, is

$$p_{sat} = 6.1078 \text{ hPa } \cdot 10^{\left(\frac{7.5T}{T+273.3}\right)}, \tag{4.1}$$

with temperature, T, given in Celsius. Once the partial pressure due to water vapor is calculated, it can be subtracted from the total recorded pressure to obtain the dry air partial pressure. Then, the air density can be estimated as

$$\rho = \frac{p_v M_v + p_d M_d}{RT}, \tag{4.2}$$

where M_v is the molar mass of water vapor, M_d the molar mass of dry air, R the ideal gas constant, and T the measured temperature in Kelvin. It is critical to recognize that air density, ρ needs to be expressed in grams per cubic centimeter for the code to give meaningful results. An example MATLAB script to convert radiosonde data to a zuvwtdp profile is given in Appendix B. Also note that the final row (high altitude) entries in the radiosonde data may be empty, particularly in the wind speed and direction columns. This is usually due to the calculation done by the radiosonde utilizing a difference method to determine wind speed and direction. These final two rows should be omitted, even if wind data is complete.

Another compatible form of atmospheric data is the U.S. Naval Research Laboratory (NRL) Ground to Space (G2S) specification[4]. The G2S atmosphere specification utilizes data from several sources including GEOS-5 data from the NASA Global Modeling and Assimilation Office (GMAO) at NASA Goddard Space Flight Center through the online data portal in the NASA Center for Climate Simulation and NOAA GFS data from the National Operational Model Archive and Distribution System (NOMADS) maintained at NOAA's National Climatic Data Center (NCDC).

Additionally, the European Centre for Medium-Range Weather Forecasts (ECMWF) has become a standard atmosphere specification source in the infrasound community. Representing efforts from 21 European Member States and 13 cooperating states, the ECMWF's system contains one of the world's largest archives of both measured and forecasted weather data. Unfortunately, public access to their data archives is not permitted without express permission.

4.3 Provided Samples

Provided with the NCPA code package are several initial signal samples and a model atmosphere file; these can be found in the `$ncpaprop/samples` directory or are accessed using builtin flags. Initial signals include a broadband pulse and templates to create your own waveform or spectrum files. Model atmosphere files include a set of profiles extracted from a Naval Research Laboratory Ground to Space semi-empirical spectral model (commonly called a "G2S" profile) template[4] and a 'toy' template featuring standard thermal profile (dip in temperature through the mesosphere, increasing temperature to the stratopause, another temperature dip in the stratosphere, and again rising temperature through the termosphere) and eastward directed wind. The initial pulse sample and the canonical "ZUVWTDP" template are depicted in Figures 4.1 and 4.2, respectively.

Figure 4.1: Provided pulse for use with ModBB; this initial wave is used automatically with the `--use_builtin_pulse` flag.

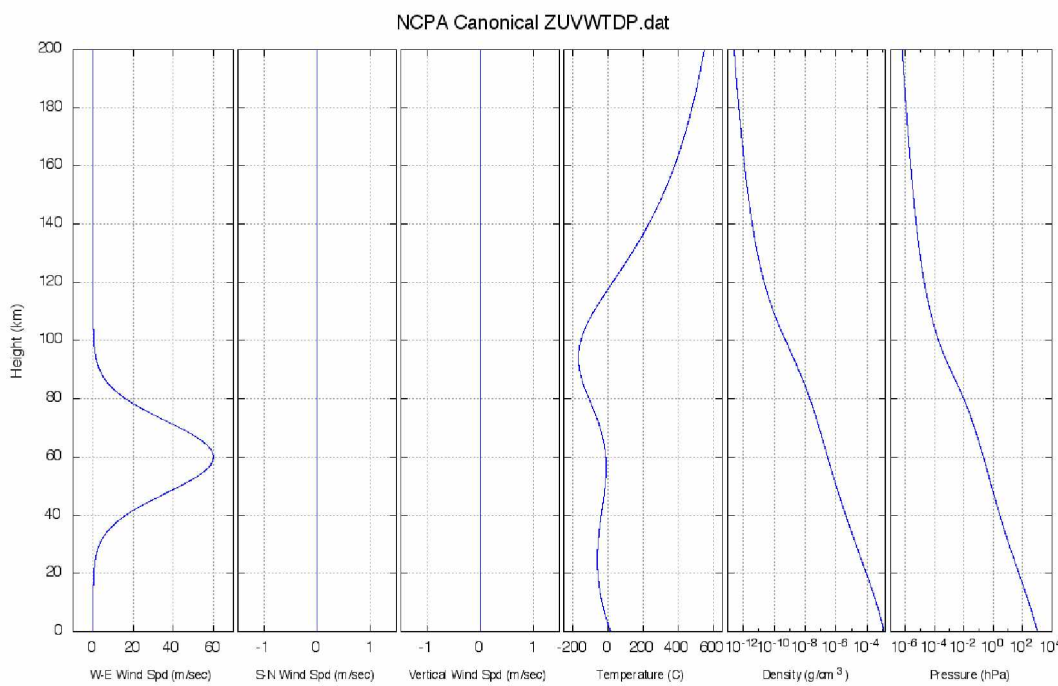

Figure 4.2: A plot of the provided NCPA Canonical ZUVWTDP sample atmospheric profile, located in the `$ncpaprop/samples` directory. Note the presence of eastward wind, standard thermal profile, and roughly exponential density and pressure profiles.

Chapter 5 2009 Sayarim Infrasound Calibration Experiment

5.1 Case Information

Three large-scale infrasound calibration experiments were conducted in 2009 and 2011 to test the International Monitoring System infrasound network and provide ground truth data for infrasonic propagation studies[6]. An investigation of the 2009 experiment is recreated here to demonstrate the use of the NCPA suite in an applied manner.

The 2009 experiment occurred on 26 August 2009 at the Sayarim Military Range in Israel (30.00057°N, 34.81351°E). A 96.0 tons equivalent of TNT explosive at 556 meters altitude (ground level) was detonated. Infrasound arrays distributed in the area detected infrasound signals out to 3500 km to the northwest, including the PELO station in southern Greece (36.90980°N, 22.47067°E) at a 1377 km range 300.3°N of Sayarim. Published propagation studies predicted infrasound in the northwest aided by a moderately strong (40 m/sec) stratospheric wind jet causing range-dependent ducting. Stratospheric and very weak thermospheric arrivals at PELO (1400 km range at 300° azimuth) are predicted according to raytracing and 0.5 Hz range dependent parabolic calculations. Predicted stratospheric arrival celerities, or recorded sound speed based on measurements, are 0.0304 km/sec (4350 second travel time) and thermospheric arrival celerities are 0.235 km/sec (5855 second travel time).

5.2 Atmosphere Model Specification

The atmospheric profile provided by Dr. Fee for the purposes of this case study was in the form of a G2S environment file. It covered the path from the Sayarim test site to the PELO station at a near 1 km range resolution. The first step was extraction of column formatted, space delimited ASCII profiles. This was done using a MATLAB script to read in the binary G2S environment file, transform the wind speed into a South-North/West-East coordinate system and calculate density, and write to individual profile files. This script is provided in Appendix 4. The atmosphere profile at zero range is plotted in Figure 5.1. Note that this atmosphere profile, and thus the effective sound speed shown in Figure 5.2, is slightly different than that used by Dr. Fee. This difference has some important consequences. Specifically, note that the effective sound speed just above 60 km (due to the stratospheric wind jet) nearly meets that at the ground.

5.3 Raytracing

A 2D raytrace was calculated for launch elevation angles from 0°to 40° and a range of 1400 km at 300° N azimuth using the `raytrace.2d` routine. The default rigid ground impedance model (full reflection) was used. The resulting raytrace plot is shown in Figure 5.3. Note that the shallowest launch angles are predicted to arrive at PELO after turning down in the downward refracting portion of the atmosphere at 40 km altitude while some

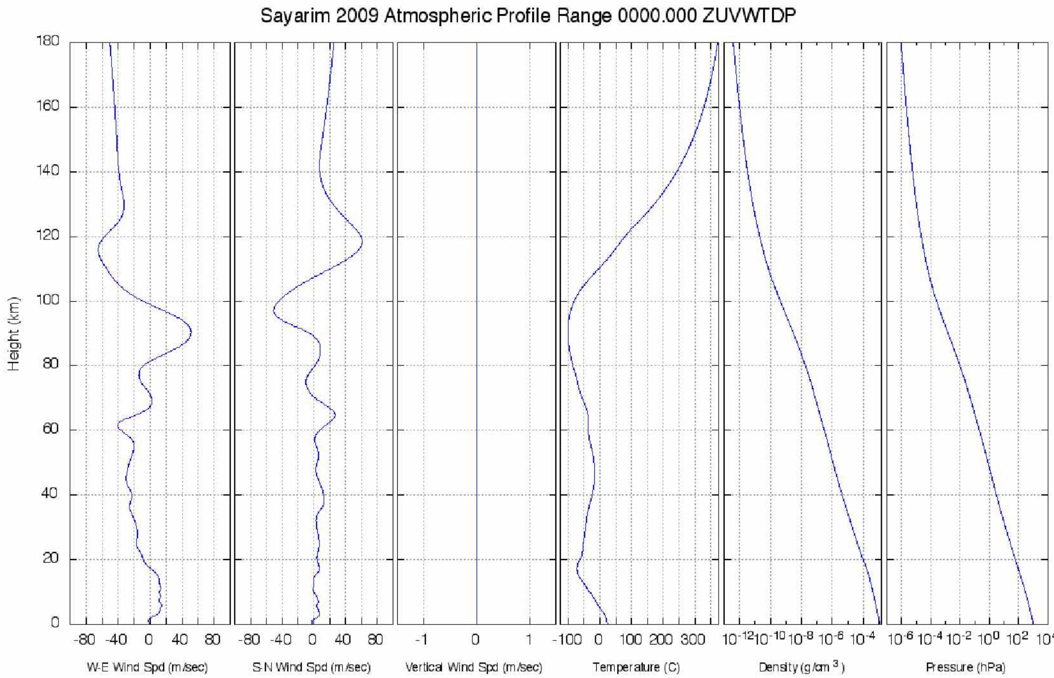

Figure 5.1: Atmosphere profile at zero range extracted from G2S Environment file along the path from Sayarim to PELO station in 2009 Calibration experiment. Note the strong winds from 80 to 130 km altitude; these wind jets persist along the path from Sayarim to PELO, facilitating ducting of infrasound in the northwest direction.

waves turn back down after reaching 60 km altitude. The remaining waves turn after reaching the thermosphere at about 110 km altitude. All ray paths are shown, though only rays which have a turning height within the stratospheric wind jet (60 km altitude) and the thermosphere (100+ km) have rays which arrive at PELO. There are also regions where very few or no rays arrive; these regions are referred to as 'shadow' regions. Transmission loss is not calculated using this raytrace routine.

5.4 Range Independent Calculations of Transmission Loss

Range independent transmission loss was calculated for 0.5 Hz signals using the `ModESS` modal routine. The zero-range atmosphere profile at Sayarim, pictured in Figure 5.1, was used for the entire propagation path. Atmospheric attenuation terms were disabled. The default rigid ground impedance model (full reflection) was used. The 1D transmission loss plot is shown in Figure 5.4 and 2D transmission loss is shown in Figure 5.5. The 1D transmission loss is compared to the expected $1/r$ decay rate, where r is the range, converted into dB. The low intensity 'shadow' regions are located below the $1/r$ line while the areas above the $1/r$ line are considered to be ducted signals. At 1400 km range, the transmission loss is just below -100 dB, consistent with the original predictions done by Dr. Fee. In contrast to the original predictions, the 2D transmission loss predicts the strongest signals

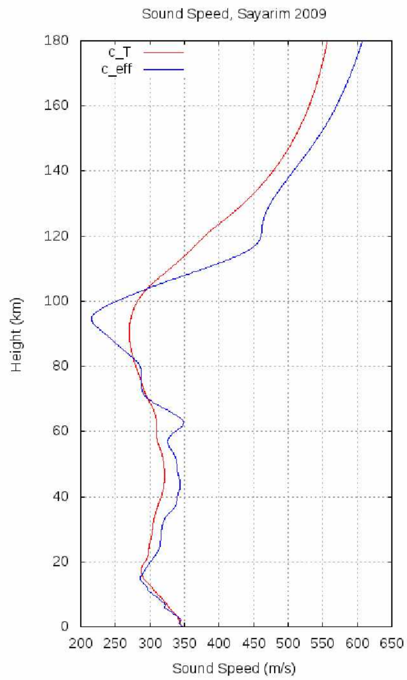

Figure 5.2: Comparison of sound speed along the path of propagation from Sayarim to PELO calculated by temperature (`c_T`, plotted in red) and effective sound speed approximation (`c_eff`, plotted in blue) during the 2009 Calibration experiment.

that arrive are those that refracted from the thermosphere between 100 and 140 km altitude while the stratospheric signals are not predicted to contribute. This is likely due to the lower effective sound speed in the stratospheric wind jet at 60 km altitude.

5.5 Range Dependent Calculations of Transmission Loss

Range dependent transmission loss was calculated for 0.5 Hz signals using the `PaPE` parabolic equation (with six coefficients) routine. The G2S environment file, which includes vertical atmosphere specifications along the propagation path from Sayrim to PELO at nearly 1 km range resolution, was used. Atmospheric attenuation terms were disabled. The default rigid ground impedance model (full reflection) was used. The 1D transmission loss plot is shown in Figure 5.6 and 2D transmission loss is shown in Figure 5.7. These results are qualitatively similar to the range independent results above. In these results the lower sound intensity 'shadow' regions are more pronounced, especially in the near field. The transmission loss to 1400 km range is still just below -100 dB, consistent with the original predictions[6]. In similar contrast, the 2D transmission loss shows only signals which refract from the thermosphere between 100 and 140 km altitude contribute to the sound intensity predicted to arrive at PELO. This is again likely due to the lower effective sound speed in the stratospheric wind jet at 60 km altitude.

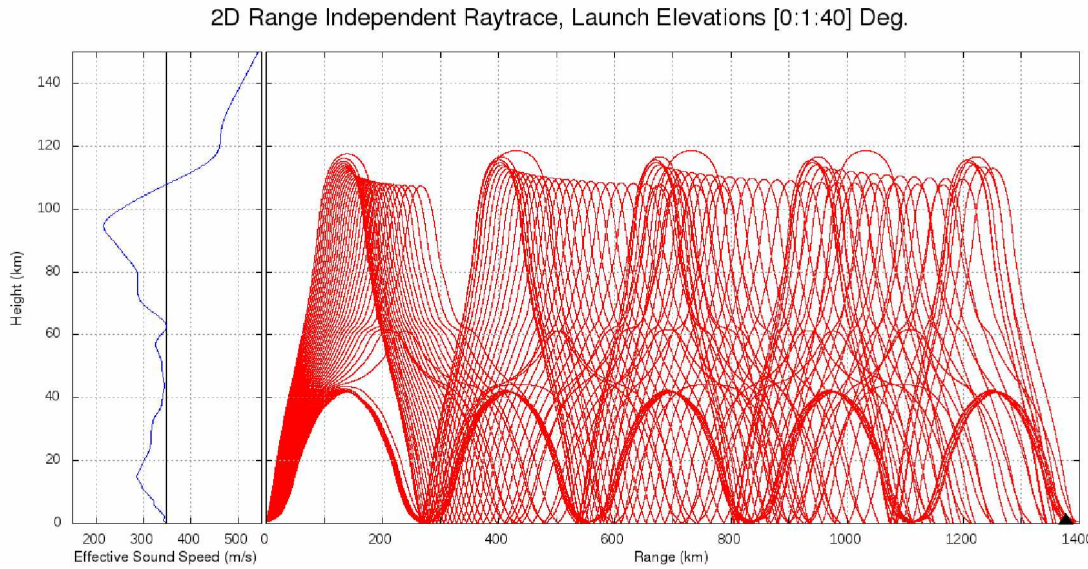

Figure 5.3: 2D range independent raytrace recreating analysis published with the 2009 Calibration experiment. The left plot is the effective sound speed in blue with a vertical line showing the effective sound speed at the ground. The right plot is the raytrace along the propagation path from Sayarim to PELO. The signal source is at the origin (bottom left) and the PELO receiving station is located at 1377 km range, indicated by a black triangle. Rays with launch angles from 0° to 40° above horizontal are shown. Note the three common ray turning altitudes influenced by the downward refracting portions of the effective sound speed at 40, 60, and 120 km altitude.

5.6 Discussion

The results predicted by the NCPA Propagation code suite are qualitatively different than those in the original predictions. Raytracing predicts the possibility the similar signal arrivals refracting from the similar atmospheric strata as in the original study. The transmission losses predicted here are coincidentally of the same magnitude as the previous study but correspond to signals refracted from the thermosphere rather than refracted from the stratospheric wind jet at about 60 km altitude. These qualitatively different results are likely caused by the lack of atmospheric absorption in these calculations; the original study used an atmospheric attenuation model. Additionally, the lack of stratospheric signal returns are likely due to the lower effective sound speed in the stratospheric wind jet at 60 km altitude, demonstrating the importance of selecting an atmospheric specification.

The routines discussed in here have more capabilities, including a macro to calculate the transmission loss at a specified range for different horizontal azimuths or to include complex eigenvalue solutions. Additionally, there are more routines than those demonstrated in this case study, including a modal routine designed to better handle higher launch angles and higher winds approaching mach speeds as well as a routine for propagating broadband sample pulses through a model atmosphere and establishing measures of dispersion.

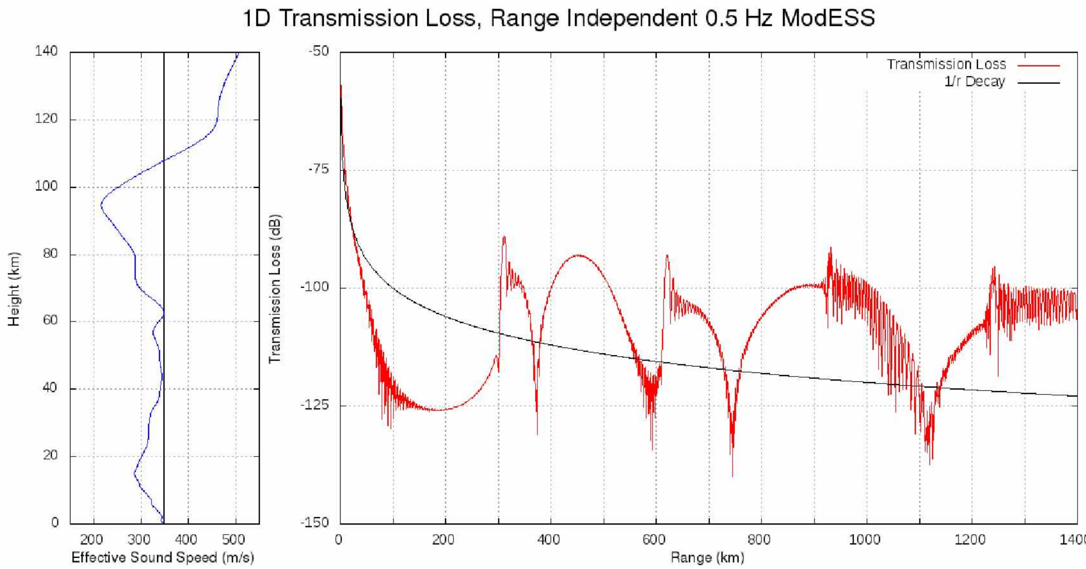

Figure 5.4: 1D range independent transmission loss calculated via 0.5 Hz Modess recreating analysis published with the 2009 Calibration experiment. The left plot is the effective sound speed in blue with a vertical line showing the effective sound speed at the ground. The right plot is the 1D transmission loss along ground in the propagation path from Sayarim to PELO compared to an expected $1/r$ decay. 'Shadow' regions are those below the $1/r$ line; regions of higher sound intensity, due to refractive ducting, are above the 1/r line.

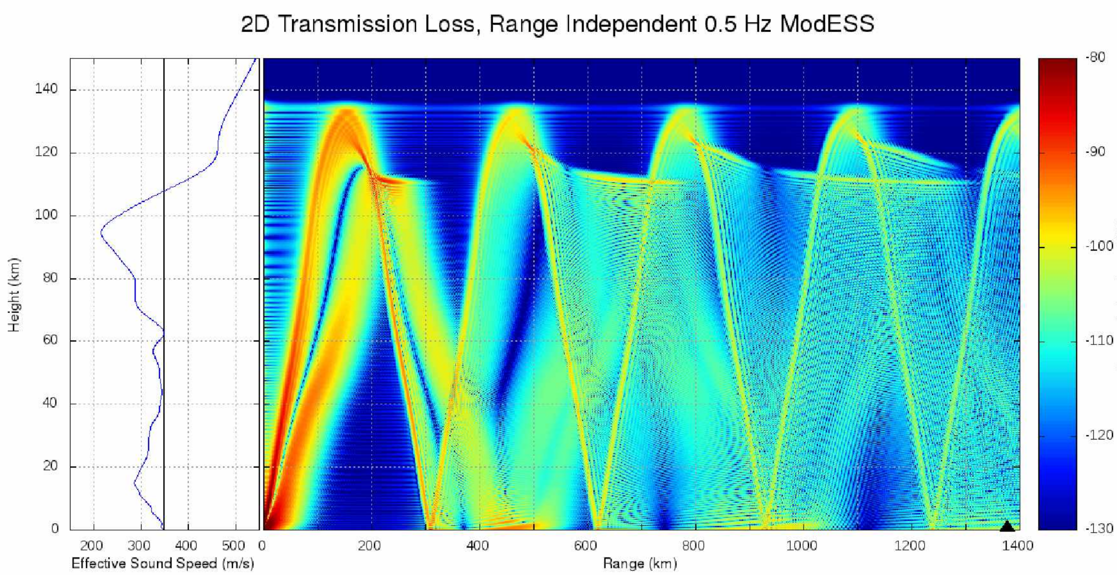

Figure 5.5: 2D range independent transmission loss calculated via 0.5 Hz Modess recreating analysis published with the 2009 Calibration experiment. The left plot is the effective sound speed in blue with a vertical line showing the effective sound speed at the ground. The right plot is a heat map of the sound pressure in the vertical 'slice' of atmosphere along the propagation path from Sayarim (bottom left corner) to PELO (black triangle in bottom right).

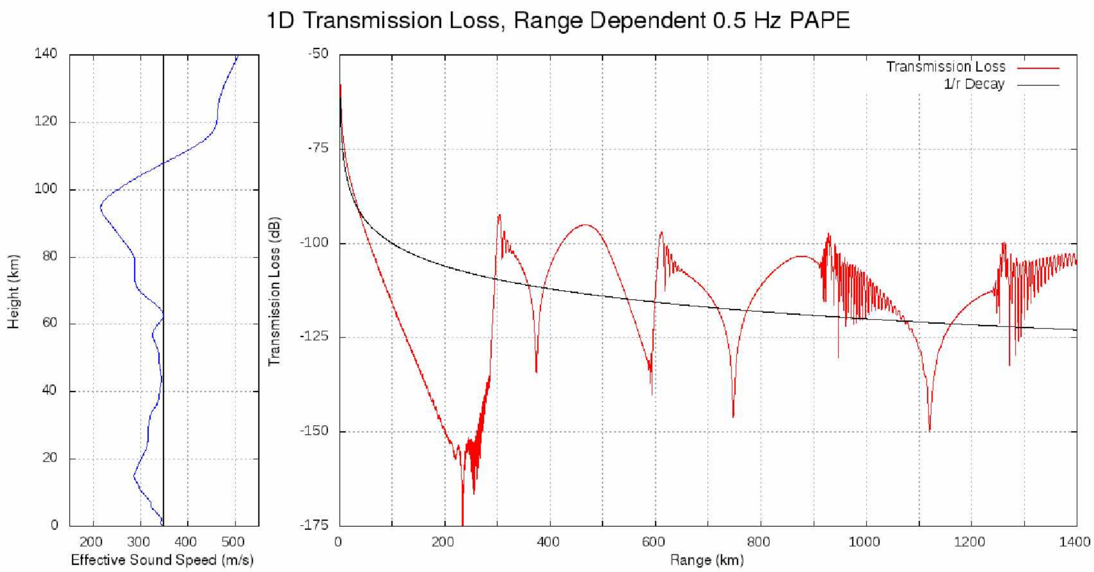

Figure 5.6: 1D range dependent transmission loss calculated via 0.5 Hz PAPE using a G2S Environment atmosphere profile recreating analysis published with the 2009 Calibration experiment. The left plot is the effective sound speed in blue with a vertical line showing the effective sound speed at the ground. The right plot is the 1D transmission loss along ground in the propagation path from Sayarim to PELO compared to an expected $1/r$ decay. 'Shadow' regions are those below the $1/r$ line; regions of higher sound intensity, due to refractive ducting, are above the 1/r line.

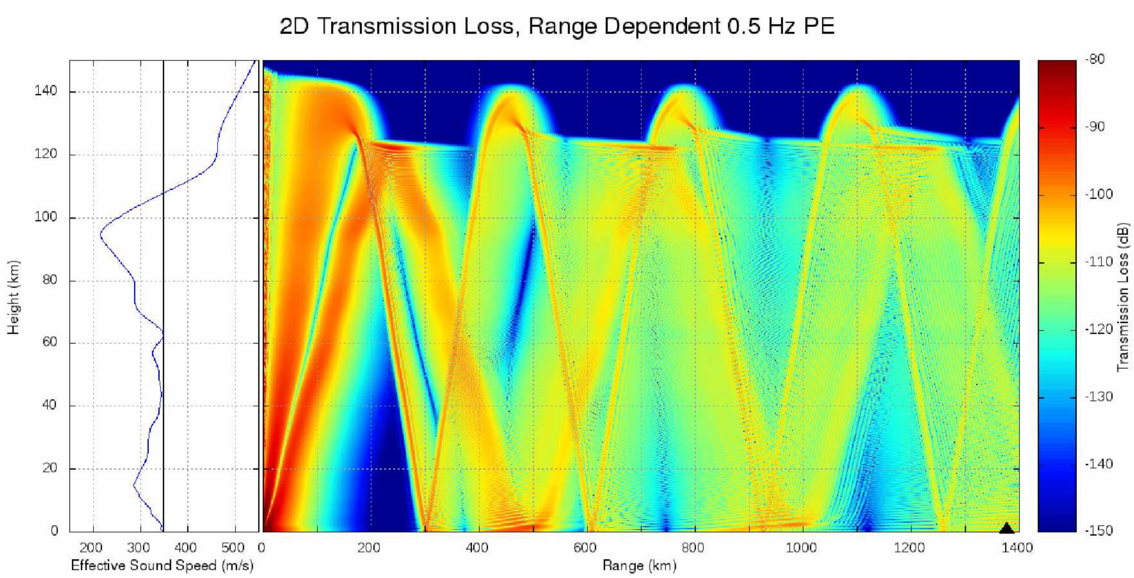

Figure 5.7: 2D range independent transmission loss calculated via 0.5 Hz PAPE using a G2S Environment atmosphere profile recreating analysis published with the 2009 Calibration experiment. The left plot is the effective sound speed in blue with a vertical line showing the effective sound speed at the ground. The right plot is a heat map of the sound pressure in the vertical 'slice' of atmosphere along the propagation path from Sayarim (bottom left corner) to PELO (black triangle in bottom right).

Chapter 6 Closing

The NCPA Propagation code suite can be a useful tool for infrasound researchers as an ‘off the shelf’ product. These routines can be utilized before time and effort is dedicated to developing a specialized code for each investigation, saving time and effort and potentially increasing productivity. In most cases, a specialized code may not be necessary as this suite fulfills nearly all propagation simulation needs. Updates for these routines will be released as ground impedance, atmospheric attenuation models, and simulation and calculations methods improve. The NCPA Propagation code suite represents the culmination of a set of ‘work in progress’ efforts to create a standard for infrasound propagation studies.

Appendix A NCPA Propagation Code Troubleshooting

A.1 Compilation Issues

A.1.1 Make returns error: petsc directory missing

If running

```
$ make
```

returns an error regarding a `petsc` or `slepc` directory missing, as

```
make: *** ncpaprop/src/extern/petsc-3.2-p7:
No such file or directory.  Stop.
make[1]: *** [.petsc-real] Error 2
make: *** [extern] Error 2
```

then ensure that line 70 in `$ncpaprop/src/Makefile` is uncommented, then run:

```
$ make clean
```

and try again.

A.1.2 Compiler complains about `uint skiplines`

If, during compilation, errors result referring to undefined type "`uint`" for the variable "skiplines," then change the type to "`int`". Do so by changing the following:

On line 16 of `$ncpaprop/src/modess_rd_1wcm/ModessRD_lib.h` change "`uint skiplines`" to "`int skiplines`"

On line 263 of `$ncpaprop/src/modess_rd_1wcm/ModessRD_lib.cpp` change "`uint skiplines`" to "`int skiplines`"

Insert at line 44 of `$ncpaprop/src/pade_pe/pe_main.cpp` a new line containing "`typedef unsigned int uint;`"

On line 24 of `$ncpaprop/src/modess_rd_2wcm/ModessRDCM_lib.h` change "`uint`" to "`std::uint`"

On line 287 of `$ncpaprop/src/modess_rd_2wcm/ModessRDCM_lib.cpp` change "`uint skiplines`" to "`int skiplines`"

Then run

```
$ make clean
```

and try again.

Appendix B Example Scripts

B.1 Testing The NCPA Propagation Routines

B.1.1 Testing 2D Raytrace

```
$ ../bin/raytrace.2d --azimuth 270 --elev 1 --delev 1 --maxelev 45 --skips 1 \
--atmosfile g2sprofile.sample --atmosfileorder ztuvpd --maxraylength 800 \
--maxheight 140 --skiplines 1
$ cat raypath_az270* > raypaths.dat
$ gnuplot -e "set output 'raypath2d_test.png'; plot `raypaths.dat' with lines"
```

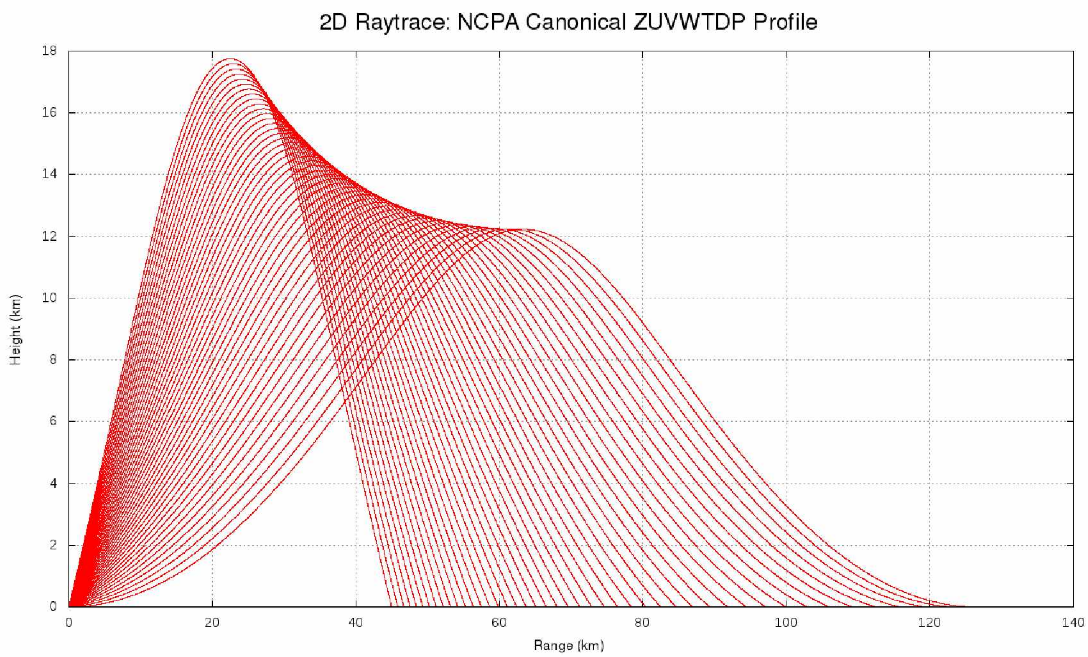

Figure B.1: Testing the 2D Raytrace routine in the provided NCPA Canonical atmosphere profile.

B.1.2 Testing 3D Raytrace

```
$ ../bin/raytrace.3d --azimuth 270 --elev 1 --delev 1 --maxelev 45 --skips 1 \
--atmosfile g2sprofile.sample --atmosfileorder ztuvpd --maxraylength 800 \
--maxheight 140 --skiplines 1
$ cat raypath_az270* > raypaths.dat
$ gnuplot -e "set output 'raypath3d_test.png'; plot `raypaths.dat' using 1:3 with lines"
$ rm raypath*
```

B.1.3 Testing ModESS

```
$ ../bin/Modess --atmosfile NCPA_canonical_profile_zuvwtdp.dat \
--atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1
```

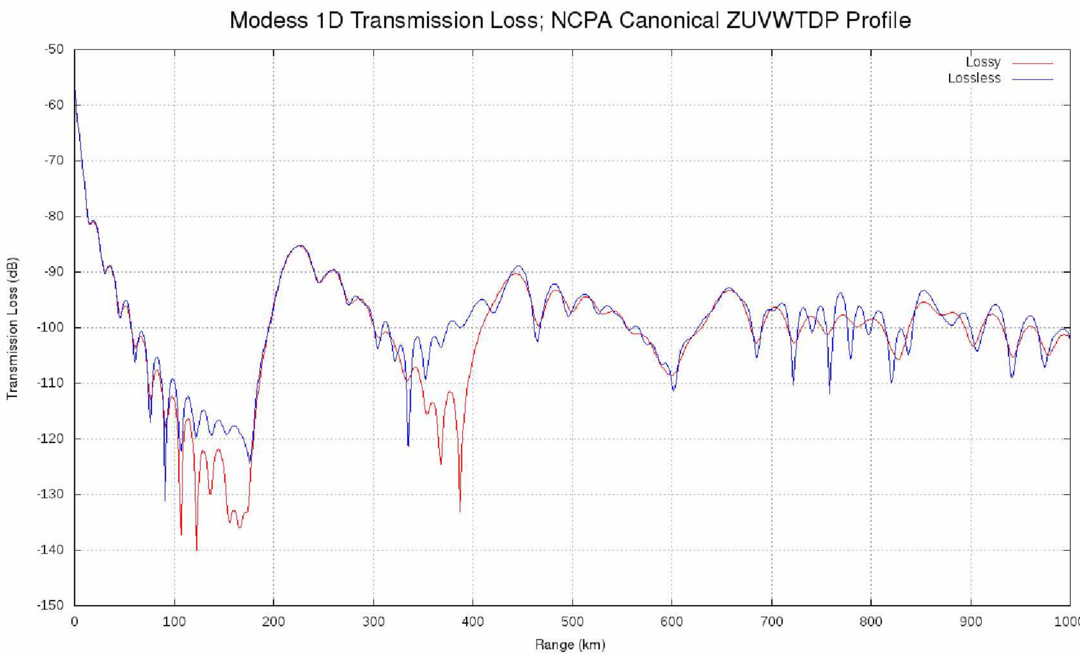

Figure B.2: Testing the Modess routine, this is the 1D transmission loss in the provided NCPA Canonical atmospheric profile

B.1.4 Testing CModESS

```
$ ../bin/CModess --atmosfile NCPA_canonical_profile_zuvwtdp.dat \
--atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1
```

B.1.5 Testing ModessRD1WCM

```
$ ../bin/ModessRD1WCM --use_1D_profiles_from_dir profiles --atmosfileorder zuvwtdp \
--skiplines 1 --azimuth 90 --freq 0.1 --use_profile_ranges_km 100_200
```

B.1.6 Testing WMod

```
$ ../bin/WMod --atmosfile NCPA_canonical_profile_zuvwtdp.dat \
--atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1
```

B.1.7 Testing ModBB

```
$ ../bin/ModBB --out_disp_src2rcv_file mydispersionfile.dat \
--atmosfile NCPA_canonical_profile_zuvwtdp.dat --atmosfileorder zuvwtdp \
--skiplines 0 --azimuth 90 --f_step 0.001953125 --f_max 0.5
$ ../bin/ModBB --pulse_prop_src2rcv mydispersionfile.dat --range_RR_km 240 \
--waveform_out_file mywavf.dat
```

B.1.8 Testing PaPE

```
$ ../bin/pape  --ncpatoy --azimuth 90 --freq 0.1 --write_2D_TLoss
```

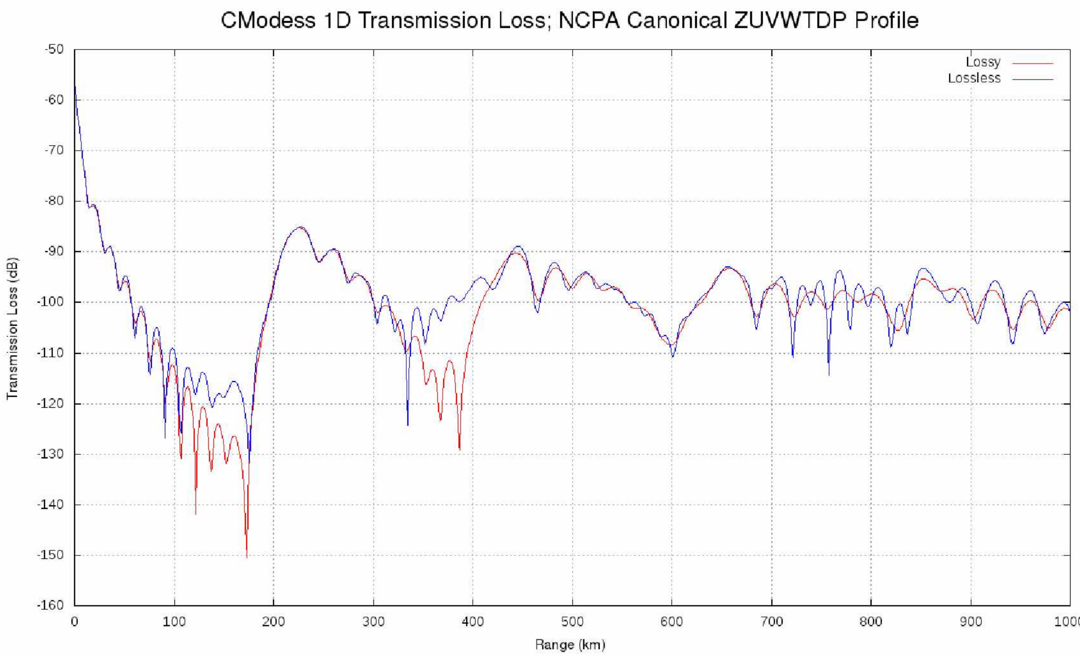

Figure B.3: Testing the CModess routine, this is the calculated 1D transmission loss in the provided NCPA Canonical atmospheric profile

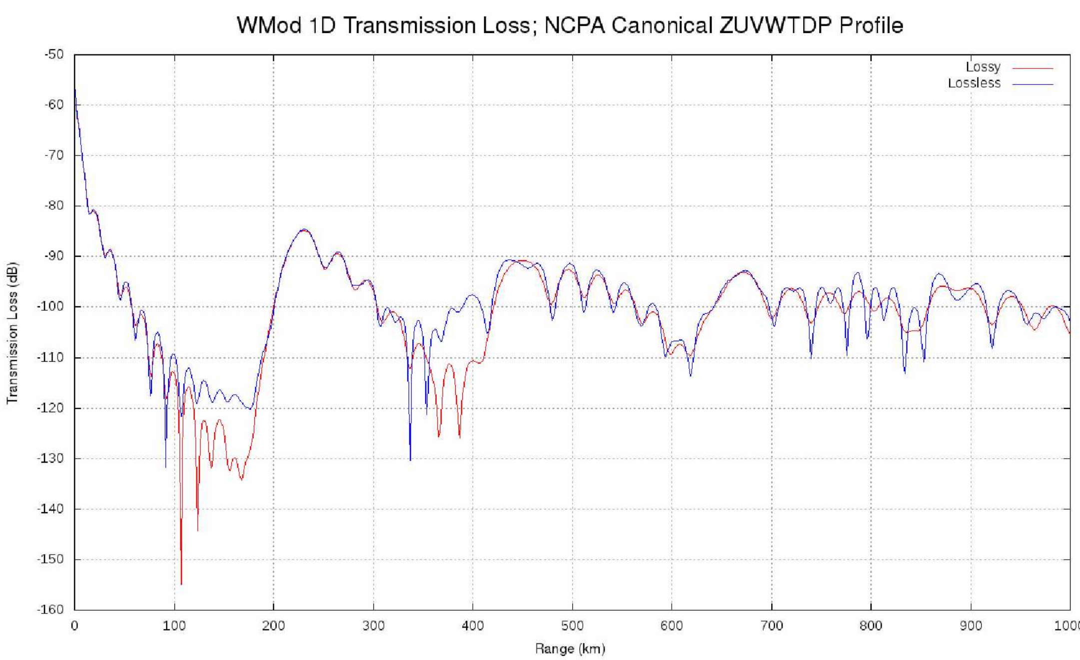

Figure B.4: Testing the WMod routine, this is the 1D transmission loss in the provided NCPA Canonical atmospheric profile

B.2 Bash Wrapper Scripts

The following scripts were used to calculate the data presented in Section 5. These are simply wrappers to call the NCPA routines with desired parameters. These are presented

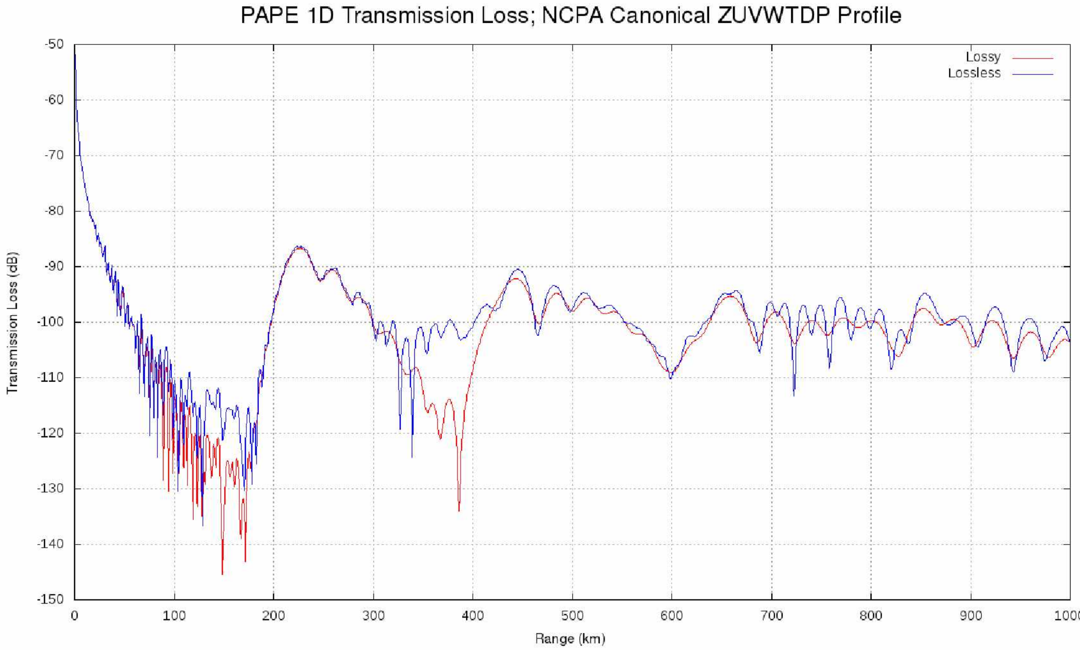

Figure B.5: Testing the PaPE routine, this is the 1D transmission loss in the provided NCPA Canonical atmospheric profile

here as reference and to demonstrate example usage of the NCPA routines. Note the redirection of output from each routine to a log file keeps a record of the routine's operation.

`run_raytrace_rangeindpendent.sh`

```
#!/bin/bash

atmfile="../atmosphere_profiles/profiles_data/profile_range_0000.000.dat"

./raytrace.2d \
    --atmosfile $atmfile \
    --atmosfileorder zuvwtdp \
    --elev 0 \
    --delev 1 \
    --maxelev 40 \
    --azimuth 300 \
    --maxrange 1400 \
    --maxheight 150 \
    --maxraylength 100000 \
    --sourceheight 0.5
```

`run_modess_rangeindpendent_2dtloss_noatmabs.sh`

```
#!/bin/bash

atmosfile='../atmosphere_profiles/profiles_data/profile_range_0000.000.dat'
atm_attn_file='../atmosphere_profiles/zero_atm_attn.dat'
```

```
./Modess \
    --atmosfile $atmosfile \
    --atmosfileorder zuvwtdp \
    --azimuth 300 \
    --freq 0.5 \
    --maxrange_km 1400 \
    --Nrng_steps 1400 \
    --sourceheight_km 0.5 \
    --skiplines 0 \
    --use_attn_file $atm_attn_file \
    --write_2D_TLoss
```

run_pape_rangedependent_envfile_2dtloss_noatmabs.sh

```
#!/bin/bash

envfile='../../atmosphere_profiles/environment/g2sgcp2009082606.env'
atm_attn_file='../../atmosphere_profiles/zero_atm_attn.dat'

./pape \
    --g2senvfile $envfile \
    --atmosfileorder zuvwtdp \
    --azimuth 300 \
    --freq 0.5 \
    --maxrange_km 1400 \
    --sourceheight_km 0.5 \
    --skiplines 0 \
    --n_pade 6 \
    --use_attn_file $atm_attn_file \
    --write_2D_TLoss \
    >> run.log
```

run_pape_rangedependent_200kmprofiles_2dtloss_noatmabs.sh

```
#!/bin/bash

profiles_1d_dir='../../atmosphere_profiles/profiles_data/d200/'
atm_attn_file='../../atmosphere_profiles/zero_atm_attn.dat'

./pape \
    --use_1D_profiles_from_dir $profiles_1d_dir \
    --atmosfileorder zuvwtdp \
    --azimuth 300 \
    --freq 0.5 \
    --maxrange_km 1400 \
    --sourceheight_km 0.5 \
    --skiplines 0 \
    --n_pade 6 \
    --use_attn_file $atm_attn_file \
    --write_2D_TLoss \
    >> run.log
```

B.3 GNUplot scripts

The following gnuplot scripts were used to generate the plots presented in Section 5 from data calculated by the wrapper scripts in Appendix 2. These are presented here as reference and to demonstrate plotting of data calculated by the NCPA routines. Note that some plotting scripts use an inline calculation in order to plot a decibel level for transmission losses. Also note that some plotting scripts such as `plot_zuvwtdp.gpl` require command line argument input.

`plot_zuvwtdp.gpl`

```
set terminal png size 1280,768 enhanced font "Helvetica,12"
# gnuplot script designed to plot the ZUVWTDP atmospheric profiles for use with the NCPA propagation code.
# call by, e.g.:
#       $ gnuplot -e "titlestr='Where ZUVWTD.dat'; outfilename='outfile.png'; infile='input.dat'" plot_zuvwtdp.gpl

# set plotting parameters, title and axes, cosmetic stuff
set output outfilename
set key off

# macros
set macros
NOYTICS = "set format y ''; unset ylabel; set ytics(''0, ''20, ''40, ''60, ''80, ''100, ''120, ''140, ''160, ''180); set yrange [0:180];"
TMARGIN2 = "set tmargin at screen 0.90; set bmargin at screen 0.1;"
TMARGIN = "set bmargin at screen 0.1;"
PLOTSTYLE = "with lines linetype 1 linewidth 1 linecolor rgb 'blue'"
# create plots
set multiplot layout 1,6 title titlestr font "Helvetica,16"

# --- u: west to east wind speed
set lmargin at screen 0.1; set rmargin at screen 0.24
set label 1 'W-E Wind Spd (m/sec)' at graph 0.5,-0.075 center font 'Helvetica,10'
set ytics ('0'0, '20'20, '40'40, '60'60, '80'80, '100'100, '120'120, '140'140, '160'160, '180'180)
set format y '%.0f'; set ylabel 'Height (km)'
set xrange [-100:100]
set xtics ('-80' -80,'' -60,'-40' -40,'' -20,'0' 0,'' 20,'40' 40,'' 60, '80' 80)
set grid
@TMARGIN
plot infile using ($2*1000):1 @PLOTSTYLE

# --- v: south to north wind speed
set lmargin at screen 0.243; set rmargin at screen 0.381
set label 1 'S-N Wind Spd (m/sec)' at graph 0.5,-0.075 center font
```

```
    ↪ 'Helvetica,10'
set xrange [-100:100]
set xtics ('-80' -80,'' -60,'-40' -40,'' -20,'0' 0,'' 20,'40' 40,'' 60, '80'
    ↪ 80)
@NOYTICS; @TMARGIN
plot infile using ($3*1000):1 @PLOTSTYLE

# --- w: vertical wind speed
set lmargin at screen 0.385; set rmargin at screen 0.523
set label 1 'Vertical Wind Spd (m/sec)' at graph 0.5,-0.075 center font
    ↪ 'Helvetica,10'
set xrange [-1.5:1.5]
set xtics ('-1' -1, '0' 0, '1' 1)
@NOYTICS; @TMARGIN
plot infile using 4:1 @PLOTSTYLE

# --- t: temperature
set lmargin at screen 0.526; set rmargin at screen 0.664
set label 1 'Temperature (C)' at graph 0.5,-0.075 center font 'Helvetica,10'
set xrange [-125:375]
set xtics ('-100' -100, '' -50, '0' 0, '' 50, '100' 100,'' 150,'200' 200,''
    ↪ 250,'300' 300,'' 350)
@NOYTICS; @TMARGIN
plot infile using ($5-273.13):1 @PLOTSTYLE

# --- d: air density
set lmargin at screen 0.668; set rmargin at screen 0.806
set label 1 'Density (g/cm^3)' at graph 0.5,-0.075 center font 'Helvetica,10'
set logscale x
set format x '10^{%T}'
set xrange [1e-13:1.5e-3]
set xtics autofreq #(''0, )
@NOYTICS; @TMARGIN
plot infile using 6:1 @PLOTSTYLE

# --- p: air pressure
set lmargin at screen 0.81; set rmargin at screen 0.95
set label 1 'Pressure (hPa)' at graph 0.5,-0.075 center font 'Helvetica,10'
set logscale x
set format x '10^{%T}'
set xrange [1e-7:1e4] #[-50:1000]
set xtics autofreq #('0' 0, '' 250, '500' 500, '' 750,'1000'1000)
@NOYTICS; @TMARGIN
plot infile using 7:1 @PLOTSTYLE
```

plot_c_t_eff.gpl

```
#!/usr/bin/gnuplot

set term png size 460,768 background rgbcolor "white"
set output "c_t_eff.png"

set key top left
set title "Sound Speed, Sayarim 2009"
set xlabel "Sound Speed (m/s)"
set ylabel "Height (km)"
set grid

plot "c_t_eff.dat" using 2:1 with lines t "c_T" linecolor rgb 'red',
   ↪ "c_t_eff.dat" using 3:1 with lines t "c_eff" linecolor rgb 'blue'
```

plot_raytrace_rangeindependent_c.gpl

```
#!/usr/bin/gnuplot

set term png size 1280,640 background rgbcolor "white"
set output "raytrace_elev_0_1_40_ceff.png"

set key off

# macros
set macros
ceff = "../atmosphere_profiles/environment/c_t_eff.dat"
TMARGIN = "set bmargin at screen 0.1;"

# get effective sound speed value at the ground
ceffg=system("head -n 1 ../atmosphere_profiles/environment/c_t_eff.dat | awk
    ↪ '{print $2}'");

# start multiplot
set multiplot layout 1,6 title "2D Range Independent Raytrace, Launch
    ↪ Elevations [0:1:40] Deg." font "Helvetica,20"

# plot c_eff
set lmargin at screen 0.075; set rmargin at screen 0.24
set label 1 'Effective Sound Speed (m/s)' at graph 0.5,-0.075 center font
    ↪ 'Helvetica,12'
set yrange [0:150]
set ytics ('0'0,'20'20,'40'40,'60'60,'80'80,'100'100,'120'120,'140'140)
set format y '%.0f'
set ylabel 'Height (km)' font 'Helvetica,12'
set xrange [150:550]
set xtics ('200'200,'300'300,'400'400,'500'500)
set grid front
set arrow from ceffg,graph(0,0) to ceffg,graph(1,1) nohead linewidth 1
    ↪ linecolor rgb 'black' #linestyle 0
@TMARGIN
plot ceff using 3:1 with lines t "c_eff" linecolor rgb 'blue'

# plot raytrace
set lmargin at screen 0.243; set rmargin at screen 0.95
set label 1 'Range (km)' at graph 0.5,-0.075 center font 'Helvetica,12'
set format y ''
unset ylabel
set yrange [0:150]
set ytics(''0,''20,''40,''60,''80,''100,''120,''140);
set xrange [0:1400]
set xtics
    ↪ ('0'0,''100,'200'200,''300,'400'400,''500,'600'600,''700,'800'800,''900,'1000'1000,''1100,'1
set grid front
```

```
unset arrow
@TMARGIN
set label 2 "" at 1377,1 point pointtype 9 pointsize 2 front
plot "raypaths_elev_0_1_40.dat" with lines linecolor rgb 'red'
```

plot_modess_rangeindependent_1dtloss_c.gpl

```
#!/usr/bin/gnuplot

set term png size 1280,640 background rgbcolor "white"
set output "tloss_1d_c.png"

# macros
set macros
ceff = "../atmosphere_profiles/environment/c_t_eff.dat"
TMARGIN = "set bmargin at screen 0.1;"

# get effective sound speed value at the ground
ceffg=system("head -n 1 ../atmosphere_profiles/environment/c_t_eff.dat | awk
    ↪ '{print $2}'");

# start multiplot
set multiplot layout 1,6 title "1D Transmission Loss, Range Independent 0.5
    ↪ Hz ModESS" font "Helvetica,20"

# plot c_eff
set lmargin at screen 0.075; set rmargin at screen 0.24
set label 1 'Effective Sound Speed (m/s)' at graph 0.5,-0.075 center font
    ↪ 'Helvetica,12'
set yrange [0:140]
set ytics ('0'0,'20'20,'40'40,'60'60,'80'80,'100'100,'120'120,'140'140)
#set format y '%.0f'
set ylabel 'Height (km)' font 'Helvetica,12'
set xrange [150:550]
set xtics ('200'200,'300'300,'400'400,'500'500)
set grid
set key off
set arrow from ceffg,graph(0,0) to ceffg,graph(1,1) nohead linewidth 1
    ↪ linecolor rgb 'black' #linestyle 0
@TMARGIN
plot ceff using 3:1 with lines t "c_eff" linecolor rgb 'blue'

# plot transmission loss
set lmargin at screen 0.31; set rmargin at screen 0.95
set label 1 'Range (km)' at graph 0.5,-0.075 center font 'Helvetica,12'
set xrange [0:1400]
set xtics
    ↪ ('0'0,''100,'200'200,''300,'400'400,''500,'600'600,''700,'800'800,''900,'1000'1000,''1100,'1
set yrange [-150:-50]
set ytics ('-150'-150,'-125'-125,'-100'-100,'-75'-75,'-50'-50)
set ylabel 'Transmission Loss (dB)' font 'Helvetica,12'
set grid
set key on
@TMARGIN
```

```
#set label 2 "" at 1377,1 point pointtype 9 pointsize 2 front
unset arrow
plot "tloss_1d.nm" using 1:(10*log10($2**2 + $3**2)) with lines linecolor rgb
    ↪ 'red' title 'Transmission Loss', "tloss_1d.nm" using
    ↪ 1:(10*log10((1/$1)**2)-60) with lines title '1/r Decay' linecolor rgb
    ↪ "black"#, "tloss_1d.nm" using 1:(10*log10($4**2)) with lines linecolor
    ↪ rgb 'green' t 'incoherent'
```

plot_modess_rangeindependent_2dtloss_c.gpl

```
#!/usr/bin/gnuplot

set term png size 1280,640 background rgbcolor "white"
set output "tloss_2d_c.png"

set key off

# macros
set macros
ceff = "../atmosphere_profiles/environment/c_t_eff.dat"
TMARGIN = "set bmargin at screen 0.1;"

# get effective sound speed value at the ground
ceffg=system("head -n 1 ../atmosphere_profiles/environment/c_t_eff.dat | awk
    ↪ '{print $2}'");

# start multiplot
set multiplot layout 1,2 title "2D Transmission Loss, Range Independent 0.5
    ↪ Hz ModESS" font "Helvetica,20"

# plot c_eff
set lmargin at screen 0.075; set rmargin at screen 0.24
set label 1 'Effective Sound Speed (m/s)' at graph 0.5,-0.075 center font
    ↪ 'Helvetica,12'
set yrange [0:150]
set ytics ('0'0,'20'20,'40'40,'60'60,'80'80,'100'100,'120'120,'140'140)
#set format y '%.0f'
set ylabel 'Height (km)' font 'Helvetica,12'
set xrange [150:550]
set xtics ('200'200,'300'300,'400'400,'500'500)
set grid front
set arrow from ceffg,graph(0,0) to ceffg,graph(1,1) nohead linewidth 1
    ↪ linecolor rgb 'black' #linestyle 0
@TMARGIN
plot ceff using 3:1 with lines t "c_eff" linecolor rgb 'blue'

# plot raytrace and transmission loss
set lmargin at screen 0.243; set rmargin at screen 0.9
set label 1 'Range (km)' at graph 0.5,-0.075 center font 'Helvetica,12'
set format y ''
unset ylabel
set yrange [0:150]
set ytics(''0,''20,''40,''60,''80,''100,''120,''140)
set xrange [0:1400]
set xtics
    ↪ ('0'0,''100,'200'200,''300,'400'400,''500,'600'600,''700,'800'800,''900,'1000'1000,''1100,'1
set cblabel "Transmission Loss (dB)" font 'Helvetica,12'
```

```
set cbrange [-130:-80]

set grid front
# matlab palette colors
set palette defined ( 0 "#000090",\
                      1 "#000fff",\
                      2 "#0090ff",\
                      3 "#0fffee",\
                      4 "#90ff70",\
                      5 "#ffee00",\
                      6 "#ff7000",\
                      7 "#ee0000",\
                      8 "#7f0000")
@TMARGIN
unset arrow
set label 2 "" at 1377,1 point pointtype 9 pointsize 2 linecolor rgbcolor
    ↪ 'black' front
set view map
plot 'tloss_2d.nm' using 1:2:(20*log10(sqrt($3**2 + $4**2))) with image#,
    ↪ '../raytrace_rangeindependent/raypaths_elev_0_1_44.dat' with lines
    ↪ linecolor rgbcolor 'black'
```

plot_pape_rangeindependent_1dtloss_c.gpl

```
#!/usr/bin/gnuplot

set term png size 1280,640 background rgbcolor "white"
set output "tloss_1d_c.png"

# macros
set macros
ceff = "../atmosphere_profiles/environment/c_t_eff.dat"
TMARGIN = "set bmargin at screen 0.1;"

# get effective sound speed value at the ground
ceffg=system("head -n 1 ../atmosphere_profiles/environment/c_t_eff.dat | awk
    ↪ '{print $2}'");

# start multiplot
set multiplot layout 1,6 title "1D Transmission Loss, Range Independent 0.5
    ↪ Hz PAPE" font "Helvetica,20"

# plot c_eff
set lmargin at screen 0.075; set rmargin at screen 0.24
set label 1 'Effective Sound Speed (m/s)' at graph 0.5,-0.075 center font
    ↪ 'Helvetica,12'
set yrange [0:140]
set ytics ('0'0,'20'20,'40'40,'60'60,'80'80,'100'100,'120'120,'140'140)
#set format y '%.0f'
set ylabel 'Height (km)' font 'Helvetica,12'
set xrange [150:550]
set xtics ('200'200,'300'300,'400'400,'500'500)
set key off
set grid
set arrow from ceffg,graph(0,0) to ceffg,graph(1,1) nohead linewidth 1
    ↪ linecolor rgb 'black' #linestyle 0
@TMARGIN
plot ceff using 3:1 with lines t "c_eff" linecolor rgb 'blue'

# plot transmission loss
set lmargin at screen 0.31; set rmargin at screen 0.95
set label 1 'Range (km)' at graph 0.5,-0.075 center font 'Helvetica,12'
set autoscale x
set xtics
    ↪ ('0'0,''100,'200'200,''300,'400'400,''500,'600'600,''700,'800'800,''900,'1000'1000,''1100,'1
set yrange [-200:-50]
set ytics
    ↪ ('-200'-200,'-175'-175,'-150'-150,'-125'-125,'-100'-100,'-75'-75,'-50'-50)
set ylabel 'Transmission Loss (dB)'
set grid
set key
```

```
@TMARGIN
unset arrow
set label 2 "" at 1377,1 point pointtype 9 pointsize 2 front
plot "tloss_1d.pe" using 1:(10*log10($2**2 + $3**2)) with lines linecolor rgb
    ↪ 'red' title 'Transmission Loss', "tloss_1d.pe" using
    ↪ 1:(10*log10((1/$1)**2)-60) with lines title '1/r Decay' linecolor rgb
    ↪ "black"
```

plot_pape_rangeindependent_2dtloss_c.gpl

```
#!/usr/bin/gnuplot

set term png size 1280,640 background rgbcolor "white"
set output "tloss_2d.png"

#set key top right
set title "2D Transmission Loss, 0.5 Hz PAPE, Sayarim 2009" font
    ↪ "Helvetica,16"
set xlabel "Range (km)" font 'Helvetica,10'
set ylabel "Height (km)" font 'Helvetica,10'
set cblabel "Transmission Loss (dB)" font 'Helvetica,10'
set yrange [0:150]
set grid

# Colorbar
# matlab palette colors
set palette defined ( 0 "#000090",\
                      1 "#000fff",\
                      2 "#0090ff",\
                      3 "#0fffee",\
                      4 "#90ff70",\
                      5 "#ffee00",\
                      6 "#ff7000",\
                      7 "#ee0000",\
                      8 "#7f0000")
set view map
set dgrid3d
#set pm3d interpolate 1000,500
set pm3d interpolate 0,0
set cbrange [-300:-50]

splot "tloss_2d.pe" u 1:2:(20*log10(sqrt($3**2 + $4**2))) with pm3d
```

B.4 MATLAB Scripts

B.4.1 MATLAB Radiosonde2zuvwtdp Atmospheric Profile Conversion

Starting with an ascii formatted radiosonde data file, e.g. `radiosonde.dat`, relevant data is calculated and formatted in columns and output to a new ascii formatted zuvwtdp data file.

`radiosonde2zuvwtdp.m`

```
function radiosonde2zuvwtdp(infile,outfile)
% radiosonde2zuvwtdp('infile','outfile')
%
% 'infile' [string] is the filename of the input radiosonde data file,
%   assumed to have 4 header lines and the file extension '.dat'
% 'outfile' [string] is the name given to the output zuvwtdp data file,
%   converted for use with the NCPA propagation code
%
% Andrew Winkelman 3 Dec 2014

% read radiosonde data from file
[pres,hght,temp,~,~,relh,~,~,drct,sknt,~,~,~] = ...
    textread(infile,'%f %f %f %f %f %f %f %f %f %f %f %f %f',...
    'headerlines',4);

% calculate needed new columns
z = hght./1000.;          % [km]
u = -sknt.*sind(drct);    % [knot]
v = -sknt.*cosd(drct);    % [knot]
w = zeros(size(z));       % [knot]
t = temp+273.3;           % [K]

% calculate air density from humidity
ps = 6.1078*10.^((7.5.*temp)./(temp+273.3));    % saturation pressure [hPa]
pv = relh/100.*ps;                              % partial pressure [hPa]
Mv = 0.018016;                                  % [kg/mol]
Md = 0.028964;                                  % [kg/mol]
R = 8.314;                                      % [J/mol*K]
d = 0.1*(pv.*Mv + (pres-pv).*Md)./(R*(t));      % [g/cm^3, NOT kg/m^3]

% write columns to output file
dlmwrite(outfile,[z u v w t d pres],'delimiter',' ','precision','%1.6e');
end
```

B.4.2 MATLAB G2S Environment Atmosphere Profile Extraction

The 2009 Sayarim to PELO Environment file provided by Fee included the MATLAB script `env_extract.m` to read the environment file to matrix form. The scripts `env_write.m` and `wind_transform.m` facilitate writing atmospheric profiles to individual column formatted, space delimited ASCII profiles.

`env_extract.m`

```
fid = fopen(filename,'r','ieee-le');

nr = fread(fid,1,'integer*4');      %[pnt] # of range point in profiles
nz = fread(fid,1,'integer*4');      %[pnt] # altitude points in profile

lat = fread(fid,nr,'real*8');       %[deg] latitude profile grid
lon = fread(fid,nr,'real*8');       %[deg] longitude profile grid
az = fread(fid,nr,'real*8');       %[deg] longitude profile grid
rng = fread(fid,nr,'real*8');       %[m] range from source of grid
alt = fread(fid,nr,'real*8');       %[m] height of air/ground/sea
ai = fread(fid,nz,'real*8');        %[km] Altitudes for 2d grid

ti = fread(fid,[nz,nr],'real*8');   %[k] Temperature
di = fread(fid,[nz,nr],'real*8');   %[g/cm3] density
pr = fread(fid,[nz,nr],'real*8');   %[hPa] pressure
ui = fread(fid,[nz,nr],'real*8');   %[m/s] Along track wind
vi = fread(fid,[nz,nr],'real*8');   %[m/s] Cross track wind
wi = fread(fid,[nz,nr],'real*8');   %[m/s] vertical wind

tj = fread(fid,[nz,nr],'real*8');   %[k] Temperature (parallel path)
dj = fread(fid,[nz,nr],'real*8');   %[g/cm3] density (parallel path)
pj = fread(fid,[nz,nr],'real*8');   %[hPa] pressure (parallel path)
uj = fread(fid,[nz,nr],'real*8');   %[m/s] Along track wind (parallel path)
vj = fread(fid,[nz,nr],'real*8');   %[m/s] Cross track wind (parallel path)
wj = fread(fid,[nz,nr],'real*8');   %[m/s] vertical wind (parallel path)
```

env_write.m

```
filename='g2sgcp2009082606.env';
env_extract

% ai is altitudes for 2d grid
% ti is temp
% di is density
% pr is pressure (avoid the constant called 'pi')
% ui is wind along propagation dirn
% vi is wind across propagation dirn
% wi is vertical wind
% baz is back-azimuth in degrees pointing back to source (seems to actually
%   be pointing in the direction of propagation) used to transform ui and vi
%   to u and v in easting and northing
% rng is range for current vertical profile, starting from source
%
% required output is columns of zuvwtdp
%
% Andrew Winkelman Aug 2015

% loop through range to produce vertical profile at each range entry:
for i=1:length(rng)
    % print status
    if i==1
        fprintf('Working on profile %04.0f of %04.0f...\n',i,length(rng))
    end
    if mod(i,10)==0
        fprintf('Working on profile %04.0f of %04.0f...\n',i,length(rng))
    end

    % transform ui and vi to easting and northing
    [u,v]=wind_transform(az(i),ui(:,i),vi(:,i));

    % write final output to file in columns, tab delimited, 1.6 float
        ↪ precision
    outfile=sprintf('../profiles_data/profile_range_%08.3f.dat',rng(i));
    dlmwrite(outfile,[ai,u./1000,v./1000,wi(:,i),ti(:,i),di(:,i),pr(:,i)],
        ↪ 'delimiter', ' ', 'precision', '%1.6e');

    if i==length(rng)
        fprintf('    Done!\n')
    end
end
```

wind_transform.m

```
function [u,v] = wind_transform(baz,ui,vi)
% [u,v] = wind_transform(baz,ui,vi)
%
% wind_transform takes back-azimuth and along track and cross track wind
% and transforms them into easting and northing wind magnitudes.
%
% Inputs:
%   baz (deg) is scaler back azimuth pointing along the direction of
    ↪ propagation
%   ui is the along track wind, in the direction of propagation
%   vi is the cross track wind, to the right of the direction of propagation
%
% Ouputs:
%   u is the easting component of wind
%   v is the northing component of wind
%
% Andrew Winkelman Aug 2015

u = -(ui.*cosd(-baz)+vi.*sind(-baz));
v = -(-ui.*sind(-baz)+vi.*cosd(-baz));
```

Appendix C NCPA Propagation Routine Help Files

Each NCPA routine includes a flag (`--help` or `-h`, for short) to display information about the usage of the routine. For convenience, these are presented here. These were generated by redirecting the command line output to a text file, e.g., by

```
$ ./raytrace.2d --help >> help_raytrace.2d.txt
```

C.1 2D Raytrace

`help_raytrace.2d.txt`

```
Usage:

The options below can be specified in a colon-separated file
    ↪ "raytrace.options" or at the command line. Command-line options
    ↪ override file options.
 --help -h                 Print this message and exit

To use an arbitrary 1-D atmospheric profile in ASCII format (space or
    ↪ comma-separated) the following options apply:
REQUIRED (no default values):
 --atmosfile <filename>   Uses an ASCII atmosphere file
 --atmosfileorder          The order of the (z,t,u,v,w,p,d) fields in the
     ↪ ASCII file (Ex: 'ztuvpd')
 --elev                    Value in range (-90,90)
 --azimuth                 Value in range [0,360), clockwise from north
 --maxraylength            Maximum ray length to calculate (km) [none]
OPTIONAL [defaults]:
 --skiplines               Lines at the beginning of the ASCII file to skip [0]
 --maxelev                 Maximum elevation angle to calculate [--elev value]
 --delev                   Elevation angle step [1]
 --maxazimuth              Maximum azimuth to calculate [--azimuth value]
 --dazimuth                Azimuth angle step [1]
 --sourceheight            Height at which to begin raytrace [ground level]
 --maxheight               Height at which to cut off calculation [150 km]
 --maxrange                Maximum distance from origin to calculate (km) [no
     ↪ maximum]
 --stepsize                Ray length step size for computation, km [0.01]
 --skips                   Maximum number of skips to allow. Use 0 for no
     ↪ limits. [0]
FLAGS (no value required):
 --partial                 Report the final, incomplete raypath as well as the
     ↪ complete bounces.

To use a set of ASCII files that form a 2-D slice of the atmosphere the
    ↪ following options apply:
REQUIRED (no default values):
 --slicefile <filename>   The name of the summary file describing the path
     ↪ (see documentation)
```

```
 --elev                   Value in range (-90,90)
 --maxraylength           Maximum ray length to calculate (km) [none]
OPTIONAL [defaults]:
 --maxelev                Maximum elevation angle to calculate [--elev value]
 --delev                  Elevation angle step [1]
 --maxazimuth             Maximum azimuth to calculate [--azimuth value]
 --dazimuth               Azimuth angle step [1]
 --sourceheight           Height at which to begin raytrace [ground level]
 --maxheight              Height at which to cut off calculation [150 km]
 --maxrange               Maximum distance from origin to calculate (km) [no
    ↪ maximum]
 --stepsize               Ray length step size for computation, km [0.01]
 --skips                  Maximum number of skips to allow.  Use 0 for no
    ↪ limits.  [0]
FLAGS (no value required):
 --partial                Report the final, incomplete raypath as well as the
    ↪ complete bounces.

To use an analytic profile with Gaussian wind jets the following options
   ↪ apply:
REQUIRED (no default values):
 --jetfile <filename>     The parameter file for the profile
 --elev                   Value in range (-90,90)
 --azimuth                Value in range [0,360), clockwise from north
    ↪ (optional for --envfile)
 --maxraylength           Maximum ray length to calculate (km) [none]
OPTIONAL [defaults]:
 --maxelev                Maximum elevation angle to calculate [--elev value]
 --delev                  Elevation angle step [1]
 --maxazimuth             Maximum azimuth to calculate [--azimuth value]
 --dazimuth               Azimuth angle step [1]
 --sourceheight           Height at which to begin raytrace [ground level]
 --maxheight              Height at which to cut off calculation [150 km]
 --maxrange               Maximum distance from origin to calculate (km) [no
    ↪ maximum]
 --stepsize               Ray length step size for computation, km [0.1]
 --skips                  Maximum number of skips to allow.  Use 0 for no
    ↪ limits.  [0]
FLAGS (no value required):
 --partial                Report the final, incomplete raypath as well as the
    ↪ complete bounces.
```

C.2 3D Raytrace

help_raytrace.3d.txt

```
Usage:

The options below can be specified in a colon-separated file
    ↪ "raytrace.options" or at the command line.  Command-line options
    ↪ override file options.
 --help -h                  Print this message and exit

To use an arbitrary 1-D atmospheric profile in ASCII format (space or
    ↪ comma-separated) the following options apply:
REQUIRED (no default values):
 --atmosfile  <filename>    Uses an ASCII atmosphere file
 --atmosfileorder           The order of the (z,t,u,v,w,p,d) fields in the
     ↪ ASCII file (Ex: 'ztuvpd')
 --elev                     Value in range (-90,90)
 --azimuth                  Value in range [0,360), clockwise from north
 --maxraylength             Maximum ray length to calculate (km) [none]
OPTIONAL [defaults]:
 --skiplines                Lines at the beginning of the ASCII file to skip [0]
 --maxelev                  Maximum elevation angle to calculate [--elev value]
 --delev                    Elevation angle step [1]
 --maxazimuth               Maximum azimuth to calculate [--azimuth value]
 --dazimuth                 Azimuth angle step [1]
 --sourceheight             Height at which to begin raytrace [ground level]
 --maxheight                Height at which to cut off calculation [150 km]
 --maxrange                 Maximum distance from origin to calculate (km) [no
     ↪ maximum]
 --stepsize                 Ray length step size for computation, km [0.01]
 --skips                    Maximum number of skips to allow.  Enter 0 for no
     ↪ maximum.  [0]
FLAGS (no values required):
 --partial                  Report the final, incomplete raypath as well as the
     ↪ complete bounces.

To use an analytic profile with Gaussian wind jets the following options
    ↪ apply:
REQUIRED (no default values):
 --jetfile  <filename>      The parameter file for the profile
 --elev                     Value in range (-90,90)
 --azimuth                  Value in range [0,360), clockwise from north
     ↪ (optional for --envfile)
 --maxraylength             Maximum ray length to calculate (km) [none]
OPTIONAL [defaults]:
 --maxelev                  Maximum elevation angle to calculate [--elev value]
 --delev                    Elevation angle step [1]
 --maxazimuth               Maximum azimuth to calculate [--azimuth value]
```

```
--dazimuth                Azimuth angle step [1]
--sourceheight            Height at which to begin raytrace [ground level]
--maxheight               Height at which to cut off calculation [150 km]
--maxrange                Maximum distance from origin to calculate (km) [no
    ↪ maximum]
--stepsize                Ray length step size for computation, km [0.1]
--skips                   Maximum number of skips to allow.  Enter 0 for no
    ↪ maximum.  [0]
FLAGS (no values required):
--partial                 Report the final, incomplete raypath as well as the
    ↪ complete bounces.
```

C.3 Modess

help_Modess.txt

```
-----------------------------------------------------------------------------
|                        NCPA Infrasound                                    |
|                          Normal Modes                                      |
|         Single Frequency - Effective Sound Speed Approximation            |
|                  Attenuation added perturbatively                         |
-----------------------------------------------------------------------------

Usage:
By default the program computes the 1D transmission loss (TL)
at the ground or the specified receiver height and saves the data to 2 files:
   file tloss_1d.nm - considering attenuation in the atmosphere
   file tloss_1d.lossless.nm  - no attenuation
Additionally, if the flag --write_2D_TLoss is present on the command line
the 2D TL is saved to file tloss2d.nm
The user can also choose to propagate in N different directions
i.e. (N by 2D mode) by using the option --Nby2Dprop .

The options below can be specified in a colon-separated file "Modess.options"
or at the command line. Command-line options override file options.
 --help -h                 Print this message and exit

To use an arbitrary 1-D atmospheric profile in ASCII format
(space or comma-separated) the following options apply:

REQUIRED (no default values):
 --atmosfile  <filename>   Uses an ASCII atmosphere file
                           referenced to Mean Sea Level (MSL).
 --atmosfileorder          The order of the (z,u,v,w,t,d,p) fields
                           in the ASCII file (Ex: 'zuvwtdp')
                           The units assumed in the ASCII file are
                           z[km], t [kelvin], d [g/cm^3], p [hectoPa]
                           The wind speeds are in m/s by default;
                           however if the winds are given in km/s then use
                           option --wind_units kmpersec
 --skiplines               Lines at the beginning of the ASCII file to skip
 --freq                    Frequency [Hz]

REQUIRED for propagation in one direction (no default values):
 --azimuth                 Degrees in range [0,360], clockwise from North

REQUIRED for propagation in N directions i.e. (N by 2D) (no default values):
 --azimuth_start           Start azimuth ([0,360] degrees, clockwise from
    ↪ North)
 --azimuth_end             End azimuth ([0,360] degrees, clockwise from North)
 --azimuth_step            Step by which the azimuth is changed (in degrees)
```

```
OPTIONAL [defaults]:
 --maxheight_km           Calculation grid height in km above MSL [150 km]
 --zground_km             Height of the ground level above MSL [0 km]
 --Nz_grid                Number of points on the z-grid from ground to
    ↪ maxheight [20000]
 --sourceheight_km        Source height in km Above Ground Level (AGL) [0]
 --receiverheight_km      Receiver height in km AGL [0]
 --maxrange_km            Maximum horizontal propagation distance from origin
    ↪ [1000 km]
 --Nrng_steps             Number of range steps to propagate [1000]
 --ground_impedance_model Name of the ground impedance models to be employed:
                          [rigid], others TBD
 --Lamb_wave_BC           If ==1 it sets admittance = -1/2*dln(rho)/dz; [ 0 ]
 --wind_units             Use it to specify 'kmpersec' if the winds are given
    ↪ in km/s [mpersec]
 --use_attn_file          Use it to specify a file name containing
    ↪ user-provided
                          attenuation coefficients to be loaded instead of
                          the default Sutherland-Bass attenuation.
                          The text file should contain two columns:
                              height (km AGL) and
                              attenuation coefficients in np/m.

FLAGS (no value required):
 --write_2D_TLoss         Outputs the 2D transmission loss to
                          default file: tloss2D.nm
 --write_phase_speeds     Output the phase speeds to
                          default file: phasespeeds.nm
 --write_modes            Output the modes to default files:
                          mode_<mode_count>.nm
 --write_dispersion       Output the mode dispersion to
                          default file: dispersion_<freq>.nm
 --Nby2Dprop              Flag to perform (N by 2D) propagation i.e.
                          propagation in N directions specified by
                          options: azimuth_start, azimuth_end, azimuth_step
                          The output is saved into default files:
                              Nby2D_tloss_1d.lossless.nm
                              Nby2D_tloss_1d.nm
 --write_atm_profile      Save the interpolated atm. profile to
                          default file: atm_profile.nm
 --turnoff_WKB            Turn off the WKB least phase speed estimation
                          an approx. that speeds-up ground-to-ground propag.
                          It has the value 1 (true) if any of the flags
                          write_2D_TLoss, write_phase_speeds, write_modes
                          or write_dispersion are true.


 The format of the output files are as follows (column order):
```

```
 tloss_1d.nm:              r, 4*PI*Re(P), 4*PI*Im(P), (incoherent TL)
 tloss_1d.lossless.nm:
 tloss_2d.nm:              r, z, 4*PI*Re(P), 4*PI*Im(P)
 Nby2D_tloss_1d.nm:        r, theta, 4*PI*Re(P), 4*PI*Im(P), (incoherent TL)
 Nby2D_tloss_1d.lossless.nm:

 phasespeeds.nm:           Mode#, phase speed [m/s], imag(k)
 mode_<mode_count>.nm      z, (Mode amplitude)
 dispersion_<freq>.nm      Contains one line with entries: freq (# of modes)
                           followed for each mode 'i' by quadruples:
                           (real(k(i)) imag(k(i)) Mode(i)(z_src) Mode(i)(z_rcv)
 atm_profile.nm            z,u,v,w,t,d,p,c,c_eff

Examples (run from 'samples' directory):
   ../bin/Modess --atmosfile NCPA_canonical_profile_zuvwtdp.dat
      ↪ --atmosfileorder zuvwtdp --azimuth 90 --freq 0.1

   ../bin/Modess --atmosfile NCPA_canonical_profile_zuvwtdp.dat
      ↪ --atmosfileorder zuvwtdp --azimuth 90 --freq 0.1 --write_2D_TLoss

   ../bin/Modess --atmosfile NCPA_canonical_profile_zuvwtdp.dat
      ↪ --atmosfileorder zuvwtdp --freq 0.1 --Nby2Dprop --azimuth_start 0
      ↪ --azimuth_end 360 --azimuth_step 1
```

C.4 CModess

help_CModess.txt

```
--------------------------------------------------------------------------------
|                          NCPA Infrasound                                     |
|                        Complex Normal Modes                                  |
|           Single Frequency - Effective Sound Speed Approximation             |
--------------------------------------------------------------------------------
Usage:
By default the program computes the 1D transmission loss (TL)
at the ground or the specified receiver height and saves the data to 2 files:
   file tloss_1d.cnm - considering attenuation in the atmosphere
   file tloss_1d.lossless.cnm  - no attenuation
Additionally, if the flag --write_2D_TLoss is present on the command line
the 2D TL is saved to file tloss2d.cnm
The user can also choose to propagate in N different directions
i.e. (N by 2D mode) by using the option --Nby2Dprop .

The options below can be specified in a colon-separated file "CModess.options"
or at the command line. Command-line options override file options.
 --help -h                Print this message and exit

To use an arbitrary 1-D atmospheric profile in ASCII format
(space or comma-separated) the following options apply:

REQUIRED (no default values):
 --atmosfile  <filename>  Uses an ASCII atmosphere file
 --atmosfileorder         The order of the (z,u,v,w,t,d,p) fields
                          in the ASCII file (Ex: 'zuvwtdp')
                          The units assumed in the ASCII file are
                          z[km], t [kelvin], d [g/cm^3], p [hectoPa]
                          The wind speeds are in m/s by default;
                          however if the winds are given in km/s then use
                          option --wind_units kmpersec
 --skiplines              Lines at the beginning of the ASCII file to skip
 --freq                   Frequency [Hz]

REQUIRED for propagation in one direction (no default values):
 --azimuth                Degrees in range [0,360], clockwise from North

REQUIRED for propagation in N directions i.e. (N by 2D) (no default values):
 --azimuth_start          Start azimuth ([0,360] degrees, clockwise from
    ↪ North)
 --azimuth_end            End azimuth ([0,360] degrees, clockwise from North)
 --azimuth_step           Step by which the azimuth is changed (in degrees)

OPTIONAL [defaults]:
 --maxheight_km           Calculation grid height in km above MSL [150 km]
```

```
--zground_km              Height of the ground level above MSL [0 km]
--Nz_grid                Number of points on the z-grid from ground to
    ↪ maxheight [20000]
--sourceheight_km        Source height in km Above Ground Level (AGL) [0]
--receiverheight_km      Receiver height in km AGL [0]
--maxrange_km            Maximum horizontal distance from origin to
    ↪ propagate [1000 km]
--Nrng_steps             Number of range steps to propagate [1000]
--ground_impedance_model Name of the ground impedance models to be employed:
                         [rigid], others TBD
--Lamb_wave_BC           If ==1 it sets admittance = -1/2*dln(rho)/dz; [ 0 ]
--wind_units             Use it to specify 'kmpersec' if the winds are given
    ↪ in km/s [mpersec]
--use_attn_file          Use it to specify a file name containing
    ↪ user-provided
                         attenuation coefficients to be loaded instead of
                         the default Sutherland-Bass attenuation.
                         The text file should contain two columns:
                             height (km AGL) and
                             attenuation coefficients in np/m.

FLAGS (no value required):
--write_2D_TLoss         Outputs the 2D transmission loss to
                         default file: tloss2D.cnm
--write_phase_speeds     Output the phase speeds to
                         default file: phasespeeds.cnm
--write_modes            Output the modes to default files:
                         mode_<mode_count>.cnm
--write_dispersion       Output the mode dispersion to
                         default file: dispersion_<freq>.cnm
--Nby2Dprop              Flag to perform (N by 2D) propagation i.e.
                         propagation in N directions specified by
                         options: azimuth_start, azimuth_end, azimuth_step
                         The output in saved into default files:
                             Nby2D_tloss_1d.lossless.cnm
                             Nby2D_tloss_1d.cnm
--write_atm_profile      Save the interpolated atm. profile to
                         default file: atm_profile.cnm
--turnoff_WKB            Turn off the WKB least phase speed estimation
                         an approx. that speeds-up ground-to-ground propag.
                         It has the value 1 (true) if any of the flags
                         write_2D_TLoss, write_phase_speeds, write_modes
                         or write_dispersion are true.


 The format of the output files are as follows (column order):
  tloss_1d.cnm:              r, 4*PI*Re(P), 4*PI*Im(P), (incoherent TL)
  tloss_1d.lossless.cnm:
```

```
 tloss_2d.cnm:                r, z, 4*PI*Re(P), 4*PI*Im(P)
 Nby2D_tloss_1d.cnm:          r, theta, 4*PI*Re(P), 4*PI*Im(P), (incoherent TL)
 Nby2D_tloss_1d.lossless.cnm:

 phasespeeds.cnm:             Mode#, phase speed [m/s], imag(k)
 mode_<mode_count>.cnm        z, (Mode amplitude)
 dispersion_<freq>.cnm        Contains one line with entries: freq (# of modes)
                              followed for each mode 'i' by (Re, Im) parts of:
                              (k(i)) Mode(i)(z_src) Mode(i)(z_rcv)
 atm_profile.cnm              z,u,v,w,t,d,p,c,c_eff

Examples (run from 'samples' directory):
   ../bin/CModess --atmosfile NCPA_canonical_profile_zuvwtdp.dat
     ↪ --atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1

   ../bin/CModess --atmosfile NCPA_canonical_profile_zuvwtdp.dat
     ↪ --atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1
     ↪ --write_2D_TLoss

   ../bin/CModess --NCPA_canonical_profile_zuvwtdp.dat --atmosfileorder
     ↪ zuvwtdp --Nby2Dprop --skiplines 0 --azimuth_start 0 --azimuth_end
     ↪ 360 --azimuth_step 1
```

C.5 WMod

help_WMod.txt

```
------------------------------------------------------------------------
|                        NCPA Infrasound                               |
|             Wide Angle High-Mach Number Normal Modes                 |
|                        Single Frequency                              |
------------------------------------------------------------------------
Usage:
By default the program computes the 1D transmission loss (TL)
at the ground and saves the data to 2 files:
   file wtloss_1d.nm - considering attenuation in the atmosphere
   file wtloss_1d.lossless.nm  - no attenuation
Additionally, if the flag --write_2D_TLoss is present on the command line the
    ↪ 2D TL is saved to file wtloss2d.nm
The user can also choose to propagate in N different directions
i.e. (N by 2D mode) by using the option --Nby2Dprop .

The options below can be specified in a colon-separated file "WMod.options"
    ↪ or at the command line.  Command-line options override file options.
 --help -h                Print these instructions and exit

To use an arbitrary 1-D atmospheric profile in ASCII format
(space or comma-separated) the following options apply:

REQUIRED (no default values):
 --atmosfile  <filename>  Uses an ASCII atmosphere file
                          referenced to Mean Sea Level (MSL).
 --atmosfileorder         The order of the (z,t,u,v,w,p,d) fields
                          in the ASCII file (Ex: 'zuvwtdp')
                          The units assumed in the ASCII file are
                          z[km], t [kelvin], d [g/cm^3], p [hectoPa]
                          The wind speeds are in m/s by default;
                          however if the winds are given in km/s then use
                          option --wind_units kmpersec
 --skiplines              Lines at the beginning of the ASCII file to skip [0]
 --azimuth                Degrees in range [0,360), clockwise from North
 --freq                   Frequency [Hz]

REQUIRED for propagation in one direction (no default values):
 --azimuth                Degrees in range [0,360], clockwise from North

REQUIRED for propagation in N directions i.e. (N by 2D) (no default values):
 --azimuth_start          Start azimuth ([0,360] degrees, clockwise from
     ↪ North)
 --azimuth_end            End azimuth ([0,360] degrees, clockwise from North)
 --azimuth_step           Step by which the azimuth is changed (in degrees)
```

```
OPTIONAL [defaults]:
 --maxheight_km             Calculation grid height in km above MSL [150 km]
 --zground_km               Height of the ground level above MSL [0 km]
 --Nz_grid                  Number of points on the z-grid from ground to
    ↪ maxheight [20000]
 --sourceheight_km          Source height in km Above Ground Level (AGL) [0]
 --receiverheight_km        Receiver height in km AGL [0]
 --maxrange_km              Maximum horizontal propagation distance from origin
    ↪ [1000 km]
 --Nrng_steps               Number of range steps to propagate [1000]
 --ground_impedance_model   Name of the ground impedance models to be employed:
                            [rigid], others TBD
 --Lamb_wave_BC             If ==1 it sets admittance = -1/2*dln(rho)/dz; [ 0 ]
 --wind_units               Use it to specify 'kmpersec' if the winds are given
    ↪ in km/s [mpersec]
 --use_attn_file            Use it to specify a file name containing
    ↪ user-provided
                            attenuation coefficients to be loaded instead of
                            the default Sutherland-Bass attenuation.
                            The text file should contain two columns:
                                height (km AGL) and
                                attenuation coefficients in np/m.

FLAGS (no value required):
 --write_2D_TLoss           Outputs the 2D transmission loss to
                            default file: wtloss2D.nm
 --write_phase_speeds       Output the phase speeds to
                            default file: wphasespeeds.nm
 --write_modes              Output the modes to default files:
                            wmode_<mode_count>.nm
 --write_dispersion         Output the mode dispersion to
                            default file: wdispersion_<freq>.nm
 --Nby2Dprop                Flag to perform (N by 2D) propagation i.e.
                            propagation in N directions specified by
                            options: azimuth_start, azimuth_end, azimuth_step
                            The output is saved into default files:
                                Nby2D_wtloss_1d.lossless.nm
                                Nby2D_wtloss_1d.nm
 --write_atm_profile        Save the interpolated atm. profile to
                            default file: atm_profile.nm
 --turnoff_WKB              Turn off the WKB least phase speed estimation
                            an approx. that speeds-up ground-to-ground propag.
                            It has the value 1 (true) if any of the flags
                            write_2D_TLoss, write_phase_speeds, write_modes
                            or write_dispersion are true.


 The format of the output files are as follows (column order):
```

```
 wtloss_1d.nm:              r, 4*PI*Re(P), 4*PI*Im(P), (incoherent TL)
 wtloss_1d.lossless.nm:
 wtloss_2d.nm:              r, z, 4*PI*Re(P), 4*PI*Im(P)
 Nby2D_wtloss_1d.nm:        r, theta, 4*PI*Re(P), 4*PI*Im(P), (incoherent TL)
 Nby2D_wtloss_1d.lossless.nm:

 wphasespeeds.nm:            Mode#, phase speed [m/s], imag(k)
 wmode_<mode_count>.nm       z, (Mode amplitude)
 wdispersion_<freq>.nm       Contains one line with entries: freq (# of modes)
                             followed for each mode i by quadruples:
                             (real(k(i)) imag(k(i)) Mode(i)(z_src)
                                ↪ Mode(i)(z_rcv)
 atm_profile.nm              z,u,v,w,t,d,p,c,c_eff

Examples (run from 'samples' directory):
   ../bin/WMod --atmosfile NCPA_canonical_profile_zuvwtdp.dat
      ↪ --atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1

   ../bin/WMod --atmosfile NCPA_canonical_profile_zuvwtdp.dat
      ↪ --atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1
      ↪ --write_2D_TLoss --sourceheight_km 60 --receiverheight_km 60

   ../bin/WMod --atmosfile NCPA_canonical_profile_zuvwtdp.dat
      ↪ --atmosfileorder zuvwtdp --freq 0.1 --Nby2Dprop --azimuth_start 0
      ↪ --azimuth_end 360 --azimuth_step 1
```

C.6 ModBB

help_ModBB.txt

```
--------------------------------------------------------------------------------
|                              NCPA Infrasound                                 |
|                          Normal Modes Broadband                              |
|     Based on either: Effective Sound Speed Approximation - see ModESS        |
|                      Wide_Angle High-Mach code - see WMod                    |
--------------------------------------------------------------------------------
Usage:

The options below can be specified in a colon-separated file "ModBB.options"
    ↪ or at the command line.  Command-line options override file options.
 --help -h                  Print this message and exit

One of two algorithms can be used to perform pulse propagation.
The first is based on the Effective Sound Speed Approximation (as in ModESS);
the second is based on the the Wide_Angle High-Mach solution
of the wave equation (see  implementation in WMod).
ModESS is faster but it is accurate for (launch) angles less than 30 deg and
low wind speeds. WMod extends the validity to higher angles
and high Mach numbers but it runs slower.
Options --use_modess and --use_wmod allow the user to choose
the desired algorithm when computing the dispersion data (see step 1 below).

To propagate a pulse, 2 steps must be completed:
 1. A dispersion file must be available or computed
     use either option --out_dispersion_files  or --out_disp_src2rcv_file
 2. Perform pulse propagation for one of several scenarios:
    a. source-to-receiver at one range (option --pulse_prop_src2rcv)
    b. source-to-receiver at several equally spaced ranges
       (option --pulse_prop_src2rcv_grid)
    c. computing the whole 2D pressure field
       (most expensive - option --pulse_prop_grid)
    For propagation the source type can be:
        delta function              -> see option --get_impulse_resp
        built-in pulse              -> see option --use_builtin_pulse
        user-provided spectrum file -> see option --src_spectrum_file
        user-provided waveform file -> see option --src_waveform_file

To compute a dispersion file: one of the following 2 options is REQUIRED:
 --out_disp_src2rcv_file  <dispersion filename>
                    Output dispersion curves and modal values for
                    source-to-receiver propagation to the specified file
 --out_dispersion_files   <dispersion filename stub>
                    Output dispersion curves and modal values on a 2D grid
                    to binary files at each frequency. The resulting filenames
                    have the stub and frequency appended:
```

```
                    e.g. <stub><freq>_nm.bin.
                    This option is computationally expensive.

 Examples (run in the 'samples' directory):

   a. Compute dispersion file that will be used to compute the pressure pulse
       ↪ at 1 receiver. Assume that we want to end up with a pulse having a
       ↪ spectrum with a maximum frequency of f_max=0.5 Hz. Also assume that
       ↪ we want the pulse represented on a time record of T=512 seconds.
       ↪ The number of positive frequencies necessary for the calculation is
       ↪ T*f_max = 256 i.e.256 frequencies between 0 and 0.5 Hz. Thus we
       ↪ know f_max=0.5 Hz and f_step=f_max/256=0.001953125 Hz. The
       ↪ corresponding run command is:

    >> ../bin/ModBB --out_disp_src2rcv_file myDispersionFile.dat --atmosfile
        ↪ NCPA_canonical_profile_zuvwtdp.dat --atmosfileorder zuvwtdp
        ↪ --skiplines 0 --azimuth 90 --f_step 0.001953125 --f_max 0.5
        ↪ --use_modess

     Each line in this dispersion file has the format:
       freq n_modes rho_src rho_rcv Re(k_pert) Im(k_pert) V_m(z_src)
           ↪ V_m(z_rcv)
     where m varies from 1 to n_modes.
     z_src and z_rcv stand for source and receiver height respectively.

   b. Compute dispersion files for propagation to all receivers on a 2D grid:
       ↪ for 256 frequencies from 0 to 0.5 Hz in steps of 0.5/256 Hz:

    >> ../bin/ModBB --out_dispersion_files disprs --atmosfile
        ↪ NCPA_canonical_profile_zuvwtdp.dat --atmosfileorder zuvwtdp
        ↪ --skiplines 0 --azimuth 90 --f_step 0.001953125 --f_max 0.5
        ↪ --use_modess

In addition the following options are REQUIRED:
 --use_modess              Prompts the use of ModESS algorithm.
 --use_wmod                Prompts the use of WMod algorithm.
 Note that --use_modess and --use_wmod are mutually exclusive.
 --atmosfile <filename>    Uses an ASCII atmosphere file
                           referenced to Mean Sea Level (MSL).
 --atmosfileorder          The order of the (z,t,u,v,w,p,d) fields in
                           the ASCII file (Ex: 'ztuvpd')
 --skiplines               Lines at the beginning of the ASCII file to skip
 --azimuth                 Value in range [0,360), clockwise from North
 --f_step                  The frequency step
 --f_max                   Maximum frequency to propagate
    Note that in this case the array of frequencies is [f_step:f_step:f_max].

OPTIONAL [defaults]:
```

```
--f_min                     Minimum frequency [f_step Hz]
--maxheight_km              Calculation grid height in km above MSL [150 km]
--zground_km                Height of the ground level above MSL [0 km]
--Nz_grid                   Number of points on the z-grid from ground to
    ↪ maxheight [20000]
--sourceheight_km           Source height in km Above Ground Level (AGL) [0]
--receiverheight_km         Receiver height in km AGL [0]
--maxrange_km               Maximum horizontal distance from origin to
    ↪ propagate
                            [1000 km]
--ground_impedance_model  Name of the ground impedance models to be employed:
                            [rigid], TBD
--Lamb_wave_BC              For a rigid ground: if ==1 it sets
                            admittance= = -1/2*dln(rho)/dz; [ 0 ]
--wind_units                Use it to specify 'kmpersec' if the winds are
    ↪ given in km/s [mpersec]

Options for PULSE PROPAGATION:
--pulse_prop_src2rcv <dispersion filename>
                   Propagate pulse from source to 1 receiver
                   at a distance specified by option --range_R_km;
--range_R_km      Propagate pulse to this range [km]
--waveform_out_file <waveform filename>   Name of the waveform output file

--pulse_prop_src2rcv_grid <dispersion filename>
                   Propagate pulse from source to array of
                   horizontally equally-spaced receivers

REQUIRED additional options:
--R_start_km       Propagation from this range to R_end_km in DR_km steps.
--R_end_km         Pulse is propagated from R_start_km to this range.
--DR_km            Range step to propagate from R_start_km to R_end_km.
--waveform_out_file <waveform filename>
                   Name of the waveform output file.

OPTIONAL [defaults]:
--f_center         The center frequency of the pulse; must be <= [f_max/5].
--max_celerity     Maximum celerity [300 m/s].



SOURCE TYPE options: Use one of the following 4 options to specify the source:
--get_impulse_resp       Flag to use a delta function as source and
                         to output the impulse response.
--use_builtin_pulse      Flag to request the use of the built-in source
    ↪ pulse.
--src_spectrum_file      Specify the file name of the source spectrum
                         at positive frequencies. The file must have 3
                             ↪ columns
```

```
                           | Freq  |  Real(Spectrum)  |  Imag(Spectrum)  |
--src_waveform_file       Specify the file name of the user-provided
                          source waveform. The file must have 2 columns
                           |Time  |  Amplitude  |
  If none of then source type options are specified the delta function source
  is the default i.e. the output is the impulse response.

Example: Pulse propagation to a point on the ground at range_R_km
         and output the impulse response:

   ../bin/ModBB --pulse_prop_src2rcv myDispersionFile.dat --range_R_km 240
      ↪ --waveform_out_file mywavf.dat --get_impulse_resp

Example: Pulse propagation to a point on the ground at range_R_km
         and employ the user-provided source spectrum:

  ../bin/ModBB --pulse_prop_src2rcv myDispersionFile.dat --range_R_km 240
     ↪ --waveform_out_file mywavf.dat --max_celerity 300
     ↪ --src_spectrum_file source_spectrum_example.dat

Example: Pulse propagation to several points on the ground 20 km apart
         and employ the user-provided source waveform:

  ../bin/ModBB --pulse_prop_src2rcv_grid myDispersionFile.dat --R_start_km
     ↪ 240 --DR_km 20 --R_end_km 300 --waveform_out_file mywavf.dat
     ↪ --src_waveform_file source_waveform_input_example.dat


To compute a 2D field:
--pulse_prop_grid <dispersion directory name>
                    Compute/view pulse on the 2D spatial x-z grid of
                        ↪ 'height_km'
                    and 'width_km' starting at 'R_start_km

  ____________________________________________________________

height_km       |                                       |
                |  Pressure field computed within      |
                |  a 2D (width_km x height_km) grid    |
                |          'ntsteps' times             |
                |      every 'tmstep' seconds          |
                |                                       |
                |                                       |
  --------------x--------------------------------------------
              R_start_km

 Additional parameters:
--R_start_km         The grid (viewing window) starts at R_start_km
--width_km           Grid width
```

```
--max_celerity      Reference speed [m/s]; in conjunction with R_start_km
                    it is determining where inside the grid the field is at
                    a time step; a value smaller than the speed of sound
                    at the ground is suggested.
--tmstep            2D pressure field is calculated at this specified time
    ↪ step.
--ntsteps           Number of times the 2D pressure field is calculated
                    'tmstep' seconds apart.

OPTIONAL [defaults]:
--height_km         The height of the 2D grid. [maximum height]
--frame_file_stub   Each 2D grid is saved into a file with the name
                    frame_file_stub_<time_of_start>; Default:[Pressure2D].

Example: >> ../bin/ModBB --pulse_prop_grid mydispersionFolder --R_start_km
    ↪ 220 --width_km 50 --height_km 25 --max_celerity 300 --tmstep 30
    ↪ --ntsteps 5 --frame_file_stub myPressure --use_builtin_pulse
```

C.7 CModBB

help_CModBB.txt

```
______________________________________________________________________________
|                              NCPA Infrasound                                |
|                     Complex Normal Modes - Broadband                        |
|          Based on the Complex Normal Mode solution (see CModess)            |
______________________________________________________________________________

Usage:

The options below can be specified in a colon-separated file "CModBB.options"
    ↪ or at the command line.  Command-line options override file options.
 --help -h                  Print this message and exit

To propagate a pulse, 2 steps must be completed:
 1. A dispersion file must be available or computed
      use either option --out_dispersion_files  or --out_disp_src2rcv_file
 2. Perform pulse propagation for one of several scenarios:
    a. source-to-receiver at one range (option --pulse_prop_src2rcv)
    b. source-to-receiver at several equally spaced ranges
       (option --pulse_prop_src2rcv_grid)
    c. computing the whole 2D pressure field
       (most expensive - option --pulse_prop_grid)
    For propagation the source type can be:
        delta function              -> see option --get_impulse_resp
        built-in pulse              -> see option --use_builtin_pulse
        user-provided spectrum file -> see option --src_spectrum_file
        user-provided waveform file -> see option --src_waveform_file

To compute a dispersion file: one of the following 2 options is REQUIRED:
 --out_disp_src2rcv_file  <dispersion filename>
                      Output dispersion curves and modal values for
                      source-to-receiver propagation to the specified file
 --out_dispersion_files   <dispersion filename stub>
                      Output dispersion curves and modal values on a 2D grid
                      to binary files at each frequency. The resulting filenames
                      have the stub and frequency appended:
                      e.g. <stub><freq>_nm.bin.
                      This option is computationally expensive.

 Examples (run in the 'samples' directory):

   a. Compute dispersion file that will be used to compute the pressure pulse
        ↪ at 1 receiver. Assume that we want to end up with a pulse having a
        ↪ spectrum with a maximum frequency of f_max=0.5 Hz. Also assume that
        ↪ we want the pulse represented on a time record of T=512 seconds.
        ↪ The number of positive frequencies necessary for the calculation is
        ↪ T*f_max = 256 i.e.256 frequencies between 0 and 0.5 Hz. Thus we
```

```
      ↪ know f_max=0.5 Hz and f_step=f_max/256=0.001953125 Hz. The
      ↪ corresponding run command is:

    >> ../bin/CModBB --out_disp_src2rcv_file myDispersionFile.dat --atmosfile
        ↪ NCPA_canonical_profile_zuvwtdp.dat --atmosfileorder zuvwtdp
        ↪ --skiplines 0 --azimuth 90 --f_step 0.001953125 --f_max 0.5

    Each line in this dispersion file has the format:
       freq n_modes rho_src rho_rcv Re(k) Im(k)
    followed by the real and imaginary parts of the modes, V_m,
       Re(V_m(z_src)) Im(V_m(z_src)) Re(V_m(z_rcv)) Im(V_m(z_rcv))
    where m varies from 1 to n_modes.
    z_src and z_rcv stand for source and receiver height respectively.

   b. Compute dispersion files for propagation to all receivers on a 2D grid:
      ↪ for 256 frequencies from 0 to 0.5 Hz in steps of 0.5/256 Hz:

    >> ../bin/CModBB --out_dispersion_files disprs --atmosfile
        ↪ NCPA_canonical_profile_zuvwtdp.dat --atmosfileorder zuvwtdp
        ↪ --skiplines 0 --azimuth 90 --f_step 0.001953125 --f_max 0.5

In addition the following options are REQUIRED:
 --atmosfile <filename>    Uses an ASCII atmosphere file
                           referenced to Mean Sea Level (MSL).
 --atmosfileorder          The order of the (z,t,u,v,w,p,d) fields in
                           the ASCII file (Ex: 'ztuvpd')
 --skiplines               Lines at the beginning of the ASCII file to skip
 --azimuth                 Value in range [0,360), clockwise from North
 --f_step                  The frequency step
 --f_max                   Maximum frequency to propagate
    Note that in this case the array of frequencies is [f_step:f_step:f_max].

OPTIONAL [defaults]:
 --f_min                   Minimum frequency [f_step Hz]
 --maxheight_km            Calculation grid height in km above MSL [150 km]
 --zground_km              Height of the ground level above MSL [0 km]
 --Nz_grid                 Number of points on the z-grid from ground to
     ↪ maxheight [20000]
 --sourceheight_km         Source height in km Above Ground Level (AGL) [0]
 --receiverheight_km       Receiver height in km AGL [0]
 --maxrange_km             Maximum horizontal distance from origin to
     ↪ propagate
                           [1000 km]
 --ground_impedance_model  Name of the ground impedance models to be employed:
                           [rigid], TBD
 --Lamb_wave_BC            For a rigid ground: if ==1 it sets
                           admittance= = -1/2*dln(rho)/dz; [ 0 ]
 --wind_units              Use it to specify 'kmpersec' if the winds are
```

```
    ↪ given in km/s [mpersec]

Options for PULSE PROPAGATION:
 --pulse_prop_src2rcv <dispersion filename>
                     Propagate pulse from source to 1 receiver
                     at a distance specified by option --range_R_km;
 --range_R_km        Propagate pulse to this range [km]
 --waveform_out_file <waveform filename>   Name of the waveform output file

 --pulse_prop_src2rcv_grid <dispersion filename>
                     Propagate pulse from source to array of
                     horizontally equally-spaced receivers

 REQUIRED additional options:
 --R_start_km        Propagation from this range to R_end_km in DR_km steps.
 --R_end_km          Pulse is propagated from R_start_km to this range.
 --DR_km             Range step to propagate from R_start_km to R_end_km.
 --waveform_out_file <waveform filename>
                     Name of the waveform output file.

 OPTIONAL [defaults]:
 --f_center          The center frequency of the pulse; must be <= [f_max/5].
 --max_celerity      Maximum celerity [300 m/s].


SOURCE TYPE options: Use one of the following 4 options to specify the source:
 --get_impulse_resp        Flag to use a delta function as source and
                           to output the impulse response.
 --use_builtin_pulse       Flag to request the use of the built-in source
    ↪ pulse.
 --src_spectrum_file       Specify the file name of the source spectrum
                           at positive frequencies. The file must have 3
                               ↪ columns
                              | Freq | Real(Spectrum) | Imag(Spectrum) |
 --src_waveform_file       Specify the file name of the user-provided
                           source waveform. The file must have 2 columns
                              |Time | Amplitude |
   If none of then source type options are specified the delta function source
   is the default i.e. the output is the impulse response.


 Example: Pulse propagation to a point on the ground at range_R_km
          and output the impulse response:

    ../bin/CModBB --pulse_prop_src2rcv myDispersionFile.dat --range_R_km 240
        ↪ --waveform_out_file mywavf.dat --get_impulse_resp

 Example: Pulse propagation to a point on the ground at range_R_km
```

```
        and employ the user-provided source spectrum:

   ../bin/CModBB --pulse_prop_src2rcv myDispersionFile.dat --range_R_km 240
      ↪ --waveform_out_file mywavf.dat --max_celerity 300
      ↪ --src_spectrum_file source_spectrum_example.dat

Example: Pulse propagation to several points on the ground 20 km apart
        and employ the user-provided source waveform:

   ../bin/CModBB --pulse_prop_src2rcv_grid myDispersionFile.dat --R_start_km
      ↪ 240 --DR_km 20 --R_end_km 300 --waveform_out_file mywavf.dat
      ↪ --src_waveform_file source_waveform_input_example.dat


To compute a 2D field:
--pulse_prop_grid <dispersion directory name>
                    Compute/view pulse on the 2D spatial x-z grid of
                        ↪ 'height_km'
                    and 'width_km' starting at 'R_start_km

  ---------------------------------------------------------------------
height_km       |                                         |
                |  Pressure field computed within         |
                |  a 2D (width_km x height_km) grid       |
                |          'ntsteps' times                |
                |      every 'tmstep' seconds             |
                |                                         |
                |                                         |
  --------------x------------------------------------------------------
             R_start_km

Additional parameters:
--R_start_km        The grid (viewing window) starts at R_start_km
--width_km          Grid width
--max_celerity      Reference speed [m/s]; in conjunction with R_start_km
                    it is determining where inside the grid the field is at
                    a time step; a value smaller than the speed of sound
                    at the ground is suggested.
--tmstep            2D pressure field is calculated at this specified time
    ↪ step.
--ntsteps           Number of times the 2D pressure field is calculated
                    'tmstep' seconds apart.
OPTIONAL [defaults]:
--height_km         The height of the 2D grid. [maximum height]
--frame_file_stub   Each 2D grid is saved into a file with the name
                    frame_file_stub_<time_of_start>; Default:[Pressure2D].

Example: >> ../bin/CModBB --pulse_prop_grid mydispersionFolder --R_start_km
```

```
↪ 220 --width_km 50 --height_km 25 --max_celerity 300 --tmstep 30
↪ --ntsteps 5 --frame_file_stub myPressure --use_builtin_pulse
```

C.8 PaPE

help_pape.txt

```
--------------------------------------------------------------------------------
|                                                                              |
|          High-Angle High-Mach Parabolic Equation  (PaPE)                     |
|                                                                              |
--------------------------------------------------------------------------------
Usage:
By default the program computes the 1D transmission loss (TL)
at the ground or the specified receiver height and saves the data to:
   file tloss_1d.pe - considering attenuation in the atmosphere
Additionally, if the flag --write_2D_TLoss is given the 2D TL is saved to
   ↪ file tloss_2d.pe
The options below can be specified in a colon-separated file "PaPE.options"
   ↪ or at the command line.  Command-line options override file options.
 --help -h                 Print this message and exit

 The atmosphere can be specified from one of 4 different sources:
    1. An .env file containing the atmospheric specifications at certain
       ↪ ranges:
       use option --g2senvfile <filename>
    2. Several ASCII files stored in a given directory:
       use option --use_1D_profiles_from_dir <mydirname>
    3. A single ASCII file. This will just force a range-independent run.
       use option --atmosfile1d <filename>
    4. A built-in NCPA canonical profile.
       use option (flag) --ncpatoy

The options available are:

REQUIRED (no default values):
 --atmosfileorder          The order of the (z,u,v,w,t,d,p) fields in the file
                           (Ex: 'ztuvpd')
 --skiplines               Lines at the beginning of the ASCII file to skip
 --azimuth                 Degrees in range [0,360), clockwise from North
 --freq                    Frequency [Hz]
 --g2senvfile <filename>   Uses an .env binary file (for range-dependent code)
 --use_1D_profiles_from_dir
                           e.g. --use_1D_profiles_from_dir <myprofiles>
                           This option allows to use the ascii profiles stored
                               ↪ in
                           the specified directory. The profiles must have
                               ↪ names
                           'profiles0001', 'profiles0002', etc. and will be
                           used in alphabetical order at the provided ranges
                           e.g. in conjunction with either
                           option  '--use_profile_ranges_km'
```

```
                      or option '--use_profiles_at_steps_km'
                      If there are more requested ranges than existing
                      profiles then the last profile is used repeatedly
                      as necessary.
   Example: >> ../bin/pape --atmosfileorder zuvwtdp --skiplines 1
            --azimuth 90 --freq 0.1 --use_1D_profiles_from_dir
               ↪ myprofiles_dir
            --use_profile_ranges_km 100_300_500_600_700

 --atmosfile1d  <filename>  Uses an ASCII 1D atmosphere file.
                      In this case the run will just be range-independent.

OPTIONAL [defaults]:
 --maxheight_km           Calculation grid height in km above MSL [150 km]
 --zground_km             Height of the ground level above MSL [0 km]
 --Nz_grid                Number of points on the z-grid from ground to
    ↪ maxheight [20000]
 --sourceheight_km        Source height in km Above Ground Level (AGL) [0]
 --receiverheight_km      Receiver height in km AGL [0]
 --maxrange_km            Maximum horiz. propagation distance from origin
    ↪ [1000 km]
 --rng_step               A usually fractional number specifying the range
    ↪ step
                          in wavelengths: e.g. --rng_step 0.1 means a step
                          of 0.1*wavelength.
 --ground_impedance_model Name of the ground impedance models to be employed:
                          [rigid], others TBD
 --n_pade                 Number of Pade coefficients [4]

 --use_profile_ranges_km
                          e.g. --use_profile_ranges_km  20_50_80.5_300
                          The profiles at certain ranges specified by numbers
                          (in km) in a string such as 20_50_80.5_300 are
                          requested in the propagation. Note that underscores
                          are necessary to separate the numbers.
                          In this example the ranges at which the profiles
                          are considered are: 0, 20, 50, 80.5, 300 km i.e.
                          0 is always the first distance even if not
                             ↪ specified.
                          Note also that these are requested ranges;
                          however the left-closest profile available
                          in the .env file will actually be used;
                          for instance we request the profile at 300 km
                          but in the .env file the left-closest profile
                          may be available at 290 km and it is the one used.
                          This convention may change in the future.
                          This option is used in conjunction with
                             --use_1D_profiles_from_dir
```

```
 --use_profiles_at_steps_km
                          e.g. --use_profiles_at_steps_km 100
                          The profiles are requested at equidistant intervals
                          specified by this option [1000]
                          This option is used in conjunction with
                              --use_1D_profiles_from_dir

 --use_attn_file          Use it to specify a file name containing
    ↪ user-provided
                          attenuation coefficients to be loaded instead of
                          the default Sutherland-Bass attenuation.
                          The text file should contain two columns:
                              height (km AGL) and
                              attenuation coefficients in np/m.


FLAGS (no value required):
 --ncpatoy                Use built-in NCPA canonical profile
 --write_2D_TLoss         Outputs the 2D transmission loss to
                          default file: tloss_2d.pe
 --do_lossless            Computation is done with no atm. absorption


 The format of the output files are as follows (column order):
  tloss_1d.pe:            r, 4*PI*Re(P), 4*PI*Im(P)
  tloss_2d.pe:         r, z, 4*PI*Re(P), 4*PI*Im(P)


-----------------------------------------------------------------------
Examples to run with various options (from the 'samples' directory):

    ../bin/pape  --ncpatoy --azimuth 90 --freq 0.1 --write_2D_TLoss

    ../bin/pape --g2senvfile g2sgcp2011012606L.jordan.env --atmosfileorder
       ↪ zuvwtdp --skiplines 0 --azimuth 90 --freq 0.3 --sourceheight_km 0
       ↪ --receiverheight_km 0 --maxheight_km 180 --starter_type gaussian
       ↪ --n_pade 6 --maxrange_km 500

    ../bin/pape --atmosfile1d NCPA_canonical_profile_zuvwtdp.dat
       ↪ --atmosfileorder zuvwtdp --skiplines 0 --azimuth 90 --freq 0.1
       ↪ --sourceheight_km 0 --receiverheight_km 0 --maxheight_km 180
       ↪ --starter_type gaussian --n_pade 4 --maxrange_km 500

    ../bin/pape --use_1D_profiles_from_dir ../samples/profiles
       ↪ --atmosfileorder zuvwtdp --skiplines 1 --azimuth 90 --freq 0.1
       ↪ --sourceheight_km 0 --receiverheight_km 0 --maxheight_km 180
       ↪ --starter_type gaussian --n_pade 6 --maxrange_km 1000
```

```
    ↪ --use_profiles_at_steps_km 20

  ../bin/pape --use_1D_profiles_from_dir ../samples/profiles
    ↪ --atmosfileorder zuvwtdp --skiplines 1 --azimuth 90 --freq 0.1
    ↪ --sourceheight_km 0 --receiverheight_km 0 --maxheight_km 180
    ↪ --starter_type gaussian --n_pade 6 --maxrange_km 1000
    ↪ --use_profile_ranges_km 0_20_60_400

______________________________________________________________________
Example to plot 1D TL with gnuplot:
 plot './tloss_1d.pe' using 1:(10*log10($2**2 + $3**2))

Example to plot 2D TL with gnuplot:
 splot './tloss_2d.pe' using 1:2:(20*log10(sqrt($3**2 + $4**2)))

______________________________________________________________________
```

References

[1] Blom, P. and Waxler, R. (2012). Impulse propagation in the nocturnal boundary layer: Analysis of the geometric component. *Journal of the Acoustical Society of America*, 131.

[2] Bodily, C. J. (2012). Shock waves in the atmosphere. *Unpublished Master's Project.*

[3] Drob, D. and Picone, J. M. (2003). Global morphology of infrasound propagation. *Journal of Geophysical Research*, 108.

[4] Drob, D. P. (2003). Detailed specifications of the atmosphere for infrasound propagation modeling. *25th Seismic Research Review.*

[5] Fee, D., Matoza, R. S., Gee, K. L., Neilsen, T. B., and Ogden, D. E. (2013a). Infrasonic crackle and supersonic jet noise from the eruption of nabro volcano, eritrea. *Geophysical Research Letters*, 40.

[6] Fee, D., Waxler, R., Assink, J., Gitterman, Y., Given, J., Coyne, J., Mialle, P., Garces, M., Drob, D., Kleinert, D., Hofstetter, R., and Grenard, P. (2013b). Overview of the 2009 and 2011 sayarim infrasound calibration experiments. *Journal of Geophysical Research: Atmospheres*, 118.

[7] Haynes, W. M. (2015). *CRC Handbook of Chemistry and Physics.* CRC Press.

[8] Hines, C. D. (1960). Internal atmospheric gravity waves at ionospheric heights. *Canadian Journal of Physics.*

[9] Jensen, F. B., Kuperman, W. A., Porter, M. B., and Schmidt, H. (2011). *Computational Ocean Acoustics, 2nd ed.* Springer.

[10] Salomons, E. M. (2001). *Computational Atmospheric Acoustics.* Kluwer Academic Publishers.

[11] Sutherland, L. C. and Bass, H. E. (2004). Atmospheric absorption in the atmosphere up to 160 km. *J. of Acoust. Soc. Am.*, 3.

[12] Waxler, R., Gilbert, K. E., and Talmadge, C. (2008). A theoretical treatment of the long range propagation of impulsive signals under strongly ducted nocturnal conditions. *Journal of the Acoustical Society of America*, 124.